

EL MICROPROCESADOR Z-80 (I)

Primitivo de FRANCISCO

En el interior del Spectrum existe un componente frecuentemente mencionado, que es el elemento inteligente capaz de interpretar y ejecutar las instrucciones, siempre en código máquina, que le proporcionamos mediante el intérprete Basic o mediante código máquina directamente. Del mayor conocimiento posible del Z-80 depende también la perfecta comprensión de futuros montajes que además, requerirán un cierto volumen de Software.

Se dice que el Z-80 es un microprocesador de la tercera generación y, en efecto, surgió como microprocesador de ocho bits que acumulaba en sí todos los perfeccionamientos del momento más los que se le añadieron.

El Z-80 sale al mercado en el año 1976, desarrollado por una nueva firma americana llamada ZILOG, y creada por un grupo de técnicos que habían trabajado anteriormente en INTEL CORPORATION en donde se había desarrollado dos años antes el famoso microprocesador 8080, que fue el primero de ocho bits que obtuvo un éxito total a escala comercial e industrial.

En estas condiciones ZILOG, con la experiencia anterior acumulada que combinaba el conocimiento del diseño y fabricación de microprocesadores con la arquitectura y software de ordenadores, y con el soporte financiero del grupo EXXON americano, creó el Z-80 cuyas siglas corresponden: Z a ZILOG y 80 porque es un digno hijo del 8080 a quien imita, compatibiliza y supera.

Sobre un chips cuya superficie es un 20% mayor que la del 8080, el Z-80 integra cerca de 8.000 transistores (Aproximadamente 4.500 más que el 8080). También hay que decir que posee 158 instrucciones (696 con los diferentes modos de direccionamiento). Tras la comercialización del Z-80 y otros, el 8080 fue quedando anticuado, permaneciendo su producción relegada a mero repuesto de los equipos ya existentes. El abandono del 8080 también fue provocado por la propia INTEL que poco después desarrolló el 8085 con no-

torias mejoras, respecto a su predecesor, el 8080.

El desarrollo de los microprocesadores de ocho bits no ha progresado posteriormente porque los fabricantes se han dedicado prioritariamente a potenciar sus nuevos microprocesadores de 16, y últimamente, 32 bits. Es por esto y por su avanzada concepción por lo que el Z-80, pese a tener casi diez años de ancianidad, sigue en cartel en el campo de los ocho bits, y hoy lo fabrican además de ZILOG: MOSTEK, SGS, NEC y SHARP.

Características del Z-80

El microprocesador Z-80 está fabricado con tecnología MOS canal N, puerta de silicio. La tecnología MOS es la más utilizada para la construcción de microprocesadores, dado que es la única que permite una gran integración a bajo costo, lo que da lugar a un reducido consumo de corriente y consecuentemente a una necesidad mínima de disipación de calor. Su capacidad de direccionamiento es de 64 Kbytes proporcionados por sus 16 bits de dirección.

La frecuencia del reloj es de 2'5 MHz y 4 MHz para la versión Z-80-A. Esta señal de reloj precisa de una única fase (Φ), lo que se traduce a un solo terminal de entrada de reloj, el cual ha de ser exterior al microprocesador.

Su encapsulado es el conocido DIL (Dual In Line) de 40 patas.

Su alimentación es únicamente de 5 voltios frente al 8080 que necesitaba además +12 V y -5 V. Posee 158 ins-



Aspecto físico del Microprocesador Z-80, cerebro del Spectrum.

trucciones en código máquina y sus modos direccionamiento son: Implícito, inmediato, relativo, directo, por registros e indexado.

Posee dos modos diferentes de interrupción no enmascarable y enmascarable, este último con tres facetas.

El Z-80 contiene un total de 22 registros de 16 y 8 bits, algunos de los de 16 bits se pueden desdoblar en dos de ocho.

Sus bus de dirección es de 16 bits y el de datos de ocho.

Su hardware externo está reducido al máximo pudiendo ser conectado a memorias RAM estáticas o dinámicas, para estas últimas dispone en su interior de una lógica completa de refresco.

Funcionamiento del Z-80

Excepto en sus particularidades ya mencionadas, el Z-80 no se diferencia sustancialmente del resto de los microprocesadores ya existentes en cuanto a su filosofía de funcionamiento; únicamente su amplia bancada de registros merece un detenido comentario.

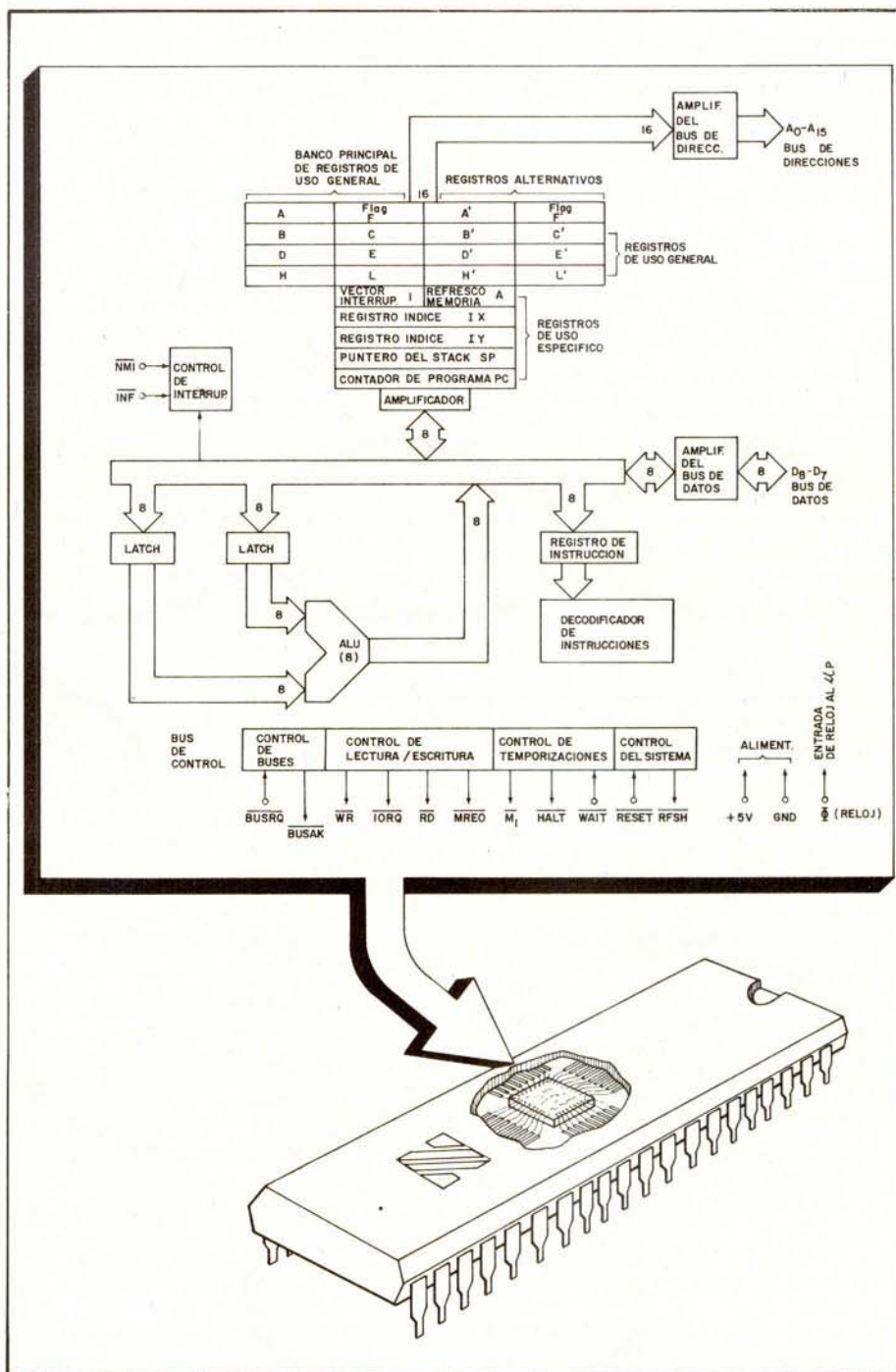
Para empezar hay que decir que toda la dinamicidad de un microprocesador depende de su entrada de reloj (Clock) que en el Z-80 es de un único terminal de entrada por donde se aplicará desde el exterior una señal de niveles TTL y a la frecuencia mencionada anteriormente. Este «clock» es el corazón del microprocesador cuyos latidos proporcionarán el ritmo de interpretación, ejecución y cálculo de las instrucciones que le llegan desde la memoria en donde han de estar contenidas previamente.

Por tanto, cuanto más alta es la frecuencia de reloj de un microprocesador, más rápido será en su ejecución,

(siempre naturalmente dentro de márgenes establecidos por el fabricante).

En el interior del Z-80 dos son los bloques encargados de interpretar y procesar las instrucciones que se le suministran ordenadamente; estos bloques son: el decodificador de instrucciones y la ALU (Siglas en inglés cuya traducción es UNIDAD ARITMETICO-LOGICA). El decodificador interpreta cada byte que le llega por el BUS de datos y activa en consecuencia los diferentes circuitos en función de una programación previa que ha introducido el fabricante en este bloque. Cada una de las posibles decodificaciones se corresponde con el primer byte del código máquina que suministra el fabricante en las tablas de instrucciones del microprocesador. Dado que el decodificador es de ocho bits, sería posible decodificar hasta 256 instrucciones; pero, puesto que en el Z-80 se ha querido disponer de hasta 696 posibilidades de interpretación, se hizo necesario ampliar a dos bytes el código de algunas instrucciones, este es el caso de aquellas cuyo primer byte es en hexadecimal: CB, DD, ED y FD. En principio parece que la codificación de instrucciones la confecciona el fabricante caprichosamente, pero si se hace un desglose de sus códigos se observa cierto orden funcional. Veamos un ejemplo: Una de las instrucciones más simples es cargar el registro "A" con un dato, su nemónico es LD A, 5D (5D es el dato). Su codificación en memoria quedaría: 3E 5D en donde 3E es el código que se corresponde con la orden: "carga el registro A con el dato 5D que viene a continuación". Un ejemplo de codificación de dos bytes sería: LD A, (IY + 07) que quiere decir: "carga el registro A con el dato que apunta el registro IY cuya dirección está formada por el contenido del registro IY + el byte que le sigue llamado desplazamiento", esta instrucción sería en código máquina FD 7E 07 en donde FD y 7E conforman los dos bytes de código y 07 es el desplazamiento.

La ALU es el bloque encargado de sumar, restar o hacer operaciones lógicas entre dos bytes y entregar el resultado por el registro A o HL afectando al registro de estado o registro de Flags denominado F. Los dos bytes a procesar han de ser previamente cargados en los LATCH de sus entradas para que estén presentes mientras opera la ALU. Todos estos procesos son contro-

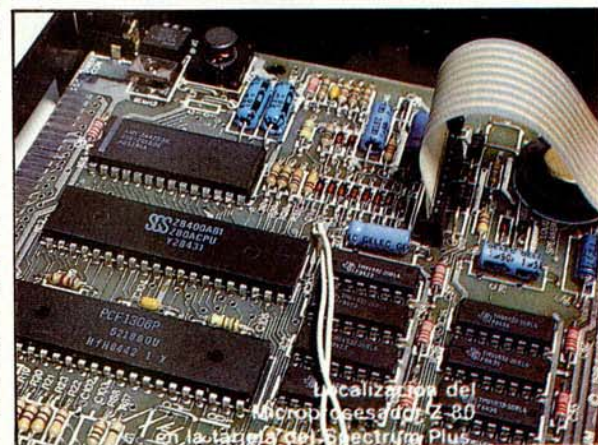


Aspecto exterior y bloques funcionales del Microprocesador Z-80.

lados por la lógica interna accionada por el decodificador de instrucciones bajo la acción del reloj, que como hemos dicho, encadena toda esta cascada de acontecimientos.

En el interior del Z-80 dos son los registros que desempeñan un papel importante en la ejecución de un programa, el contador de programa (PC) y el registro de estado (F). El contador de programa es de 16 Bits y tiene, por tanto, la capacidad de acceder a todos los bytes de la memoria que serán un total de 65.536 (en hexadecimal desde 0000 hasta FFFF). Este registro lleva

Localización del Z-80 en el interior del Spectrum Plus.



EL MICROPROCESADOR Z-80 (II)

Primitivo de FRANCISCO

El microprocesador Z-80 cobija bajo su negro caparazón, en la diminuta pastilla de silicio, un buen montón de conceptos y filosofías funcionales, fruto de años de inventiva y evolución en el mundo de los microcircuitos y del Software. En esta segunda parte proseguimos con su descripción para facilitar la comprensión del hardware que lo rodea.

El Z-80 se comunica con el exterior por 40 terminales los cuales se reparten la totalidad de las funciones operativas del microprocesador. Todos ellos funcionan con los niveles lógicos TTL porque es ésta la familia de circuitos integrados más divulgada, la cual, además ha conseguido el mayor alto grado de perfección, tanto en velocidad de conmutación como en consumo eléctrico a precios económicos. Los niveles lógicos de tensión en TTL son: de 0V hasta 0.8V para el cero lógico y desde 2,4 hasta 5V para el uno lógico.

Las salidas del Z-80 acceden hacia el exterior a través de buffers que amplifican y proporcionan a las mismas la potencia para dar señal a varias entradas TTL que se pueden conectar en paralelo. El número de éstas depende de la variante TTL que se le conecte. Frecuentemente se usa la TTL-LS, la más avanzada, que posee un menor FAN-In o corriente de entrada. Como se puede observar en todos nuestros montajes es ésta la que utilizamos, perfectamente reconocible por las siglas LS que se encuentran en medio del código comercial, por ejemplo: el circuito 74240 de TTL se convierte en 74LS240 en TTL-LS.

Como características eléctricas del Z-80 hay que añadir que es capaz de disipar hasta 1,5 vatios, que soporta unas temperaturas de almacenamiento entre -65°C y 150°C y que todas sus entradas admiten un margen de tensión desde -0,3V hasta 7V respecto a tierra (entre los cuales se encuentran, naturalmente, los niveles mencionados TTL). La descripción funcional de cada uno de sus terminales es: (Vease figura uno).

A0 - A15

Es el bus de direcciones, que sale hacia el exterior mediante 16 terminales.

D0 - D7

Es el bus de datos, se comunica con el exterior mediante 8 terminales, el buffer interior es triestado para permitir el trasiego de información tanto hacia el Z-80 como desde el Z-80.

WR

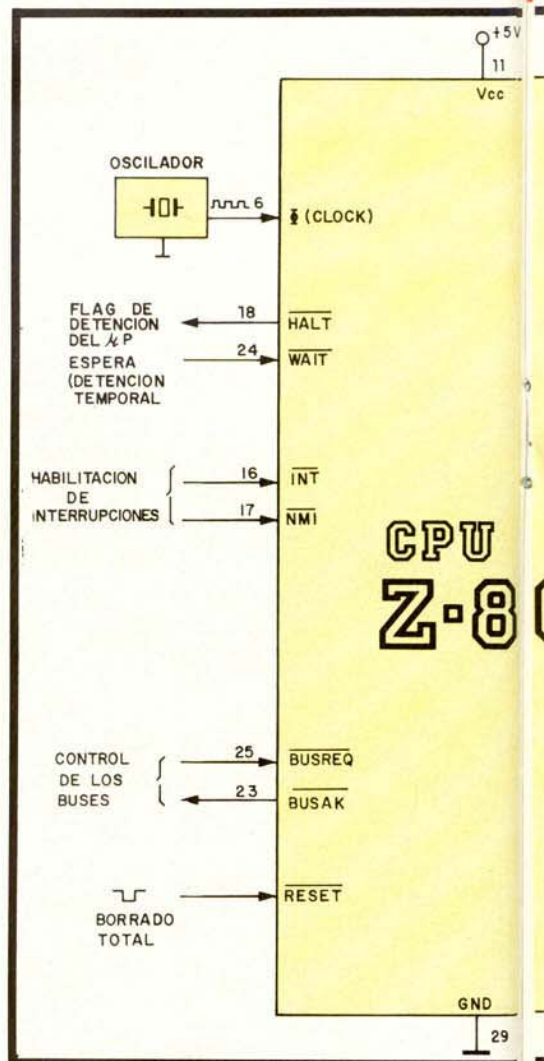
Sale por el pin 22. Cuando el Z-80 pone este terminal a cero lógico o nivel bajo indica a los periféricos o a la memoria que en este momento se va a producir una operación de escritura (Write). Esto implica que el bus de datos se convierte en bus de salida para entregar por sus ocho bits el byte adecuado.

RD

Sale por pin 21. Cuando el Z-80 lo pone a nivel bajo quiere indicar a la memoria y periféricos que en este momento se produce un ciclo de lectura (Read) y el bus de datos se pone en condiciones de recibir un byte. (Nunca se podrán dar RD y WR al mismo tiempo).

MREQ

Sale por el pin 19. El Z-80 pone este terminal a nivel bajo para indicar que la



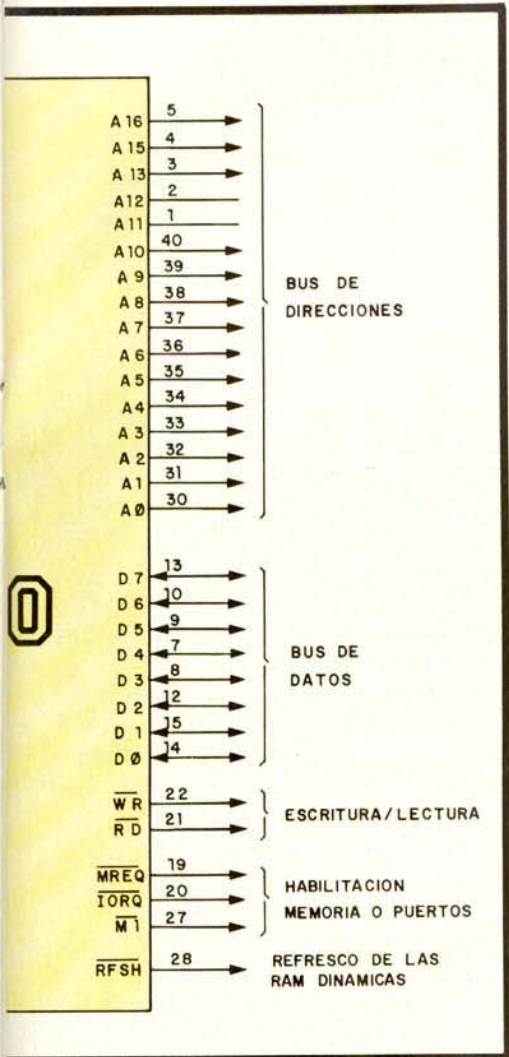
Descripción de los terminales del microprocesador Z-80.

operación que se inicia se refiere a la memoria, a la cual tiene que acceder para leer o escribir (MREQ = MEMORY REQUEST).

Esta señal se producirá lógicamente en combinación con las señales RD o WR y con el bus de direcciones y de datos para acceder a una de las 65536 direcciones posibles de memoria (RAM o ROM).

IORQ

Sale por el pin 20. El microprocesador pone a cero este terminal para indicar a los periféricos que ha de acceder alguno de ellos para leer o escribir según la dirección que presenta al tiempo por el bus de direcciones con el dato a transferir por el bus de datos. (IORQ = INPUT/OUTPUT REQUEST). El Z-80 posee también 65536 direcciones de periféricos llamadas puertos («ports»), sólo que el acceso a los puertos altos es



a nivel bajo mientras se esté procesando el byte de código de toda instrucción. M1 también pasa a nivel bajo con la señal IORQ durante el ciclo de reconocimiento de alguna interrupción. (M1 = MACHINE CYCLE ONE).

RFSH

Sale por el pin 28. Este terminal pasa a nivel bajo para que en el ciclo de refresco de las memorias RAM dinámicas, que se conecten al Z-80, la dirección de refresco salga hacia el exterior por los siete bits de dirección (A0 a A6). Esta señal se produce en combinación con MERQ que hace de clock de refresco. A7 permanece a cero y A8 a A15 presentan el contenido del registro interno I. El proceso de refrescos es complejo y de él trataremos más adelante.

Vcc

Entrada de alimentación (+ 5V), pin 11. Sobre esta entrada el Z-80 presenta el consumo de corriente hacia la fuente de 5V, esta corriente es próxima a 200 mA.

GND

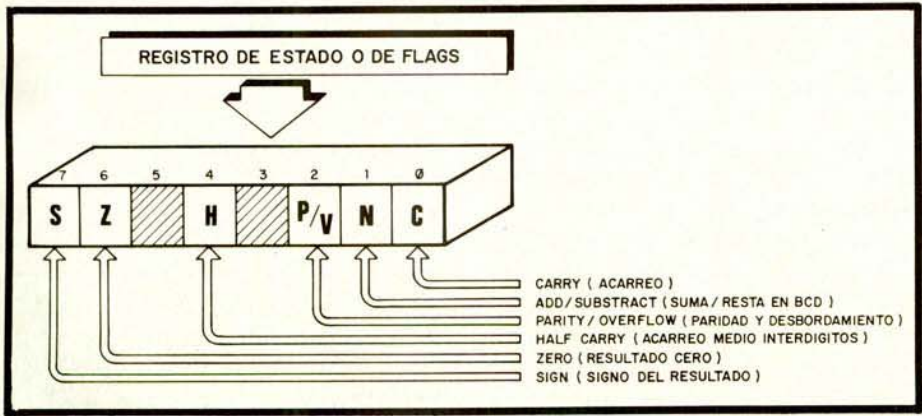
Es la conexión a tierra (Ground). Sale por el terminal 29.

CLOCK

Esta señal también se suele representar por la letra griega Φ para indicar entrada de reloj de una única fase. Se corresponde con el pin 6 y por ella entra la señal de reloj (que en el Spectrum

M1

Sale por el pin 27. Este terminal pasa



Significado de los bits del registro de estado o de flags (los bits 5 y 8 tienen sentido sólo para las operaciones internas del microprocesador y difícilmente para el usuario).

es de 3,5 Mhz), que le proporciona la ULA (500 Mhz menos de los que podría soportar el Z-80 A que es el que equipa al Spectrum).

HALT

Señal de salida por el pin 18; cuando el Z-80 la pone a nivel bajo indica al exterior que se ha producido una instrucción HALT y que, hasta que no sea sacado de este estado mediante un reset o una interrupción, el Z-80 mantendrá el refresco de memoria ejecutando constantemente instrucciones NOP (no operación).

WAIT

Entra por el pin 24; cuando algún dispositivo exterior lleva a este terminal a nivel bajo, el Z-80 permanecerá inactivo mientras dure la señal WAIT, manteniendo intactos los registros y el contenido de los buses, pero cesando el refresco de las memorias dinámicas. Es por esto que en el Spectrum esta señal no debe durar más de 2 milisegundos. Se puede usar para sincronizar al Z-80 con periféricos lentos que tienen un tiempo de conmutación superior al del Z-80. (Wait = Espera).

INT

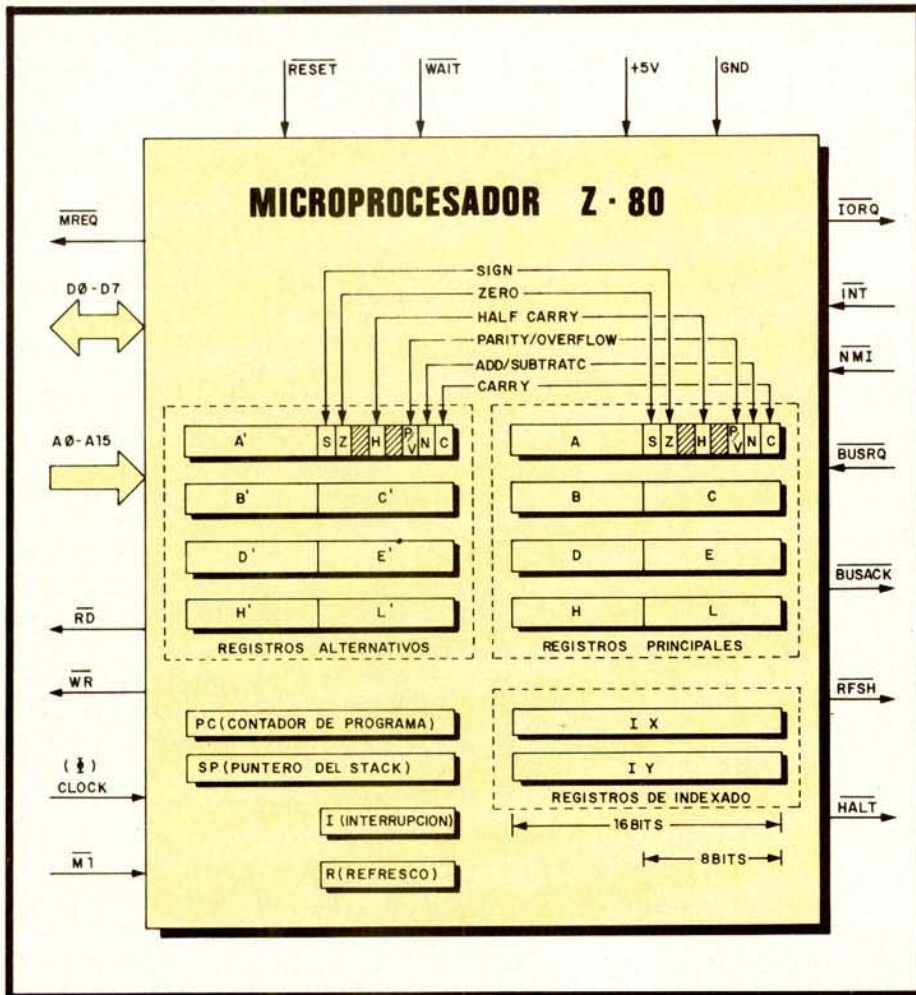
Pin 16; es la entrada de las interrupciones enmascarables.

NMI

Entra por el pin 17 y habilita las interrupciones no enmascarables. (NMI = NOM MASKABLE INTERRUPT). De las señales INT y NMI trataremos profundamente en un próximo apartado hardware.

BUSREQ

Entra por el pin 25. Es puesta a nivel bajo por algún periférico que tiene que compartir los buses del Z-80. Cuando un periférico hace petición de acceso a los buses, el Z-80 pasa a atenderlo tan pronto como concluya el ciclo máquina en ejecución e inmediatamente pone sus buses en alta impedancia para



Registros del microprocesador Z-80 y funciones de sus conexiones con el exterior.

permitir entrar en ellos a otro periférico o microprocesador. (BUSREQ = BUS REQUEST).

BUSACK

Es una salida por donde el Z-80 le comunica al periférico que solicitó por Busreq acceso a los buses, que su petición pasa a ser atendida. Para ello pone esta salida a nivel bajo. Y a alta impedancia sus buses, los cuales llevan, como ya se ha dicho, circuitos amplificadores de salida tipo triestado.

Mientras que el Z-80 se encuentra en aceptación de buses compartidos abandona el refresco de las memorias RAM DINAMICAS que se encuentran colgadas de él. Esta condición ha de estar prevista necesariamente en el desarrollo hardware que use esta opción. (Busak = bus acknowledge)

Esta salida se efectúa con el pin 25 del microprocesador.

RESET

Es una entrada al Z-80 por el pin 26. Mediante ella se consigue inicializar en todo momento al microprocesador, llevando esta entrada a nivel lógico bajo. Cuando esto ocurre todos los registros internos son puestos a cero, incluso el registro contador de programa. Al llevar a uno este terminal inmediatamente el Z-80 comienza a ejecutar lo que se halle a partir de la dirección 0000 H.

Durante el reset se producen también algunas otras condiciones que hay que conocer:

- Habilitación de las interrupciones.
- El registro de refresco R se pone también a cero iniciándose a partir de este momento un nuevo ciclo.
- El registro I que vectoriza un modo de interrupciones también se pone a cero.
- Con el reset se selecciona automáticamente el modo 0 de interrupción (IM0).

Mientras que esta entrada está a nivel bajo, todos los buses se ponen a alta

impedancia y cesa el refresco de las memorias RAM. Al mismo tiempo, el resto de las señales se ponen en estado inactivo. El reset funciona por nivel, no por flanco, por lo cual este estado puede prolongarse indefinidamente siempre que el reset esté a nivel lógico cero.

Registros del Z-80

Un registro no es más que una cadena de biestables tipo D que pueden almacenar en paralelo una información binaria temporalmente. Su uso se hace imprescindible para toda la operatividad interna del microprocesador, para la comunicación con el exterior y para el almacenamiento temporal que sirve de gran auxilio al programador.

Ya mencionamos brevemente los nombres de la totalidad de los registros; ahora los analizaremos detenidamente.

Existen dos bancos de registros idénticos compuestos ambos de cuatro registros de 16 bits. Estos dos bancos reciben el nombre de registros principales y registros alternativos.

Cada uno de estos registros pueden ser usados como registros de ocho bits o de dieciseis; para ello basta con usar las instrucciones apropiadas que los tratan como simples o como dobles. Así por ejemplo, el registro BC puede ser usado como B, como C o como BC, esta última configuración posibilita alcanzar directamente la totalidad del espacio direccionable (64 K) y operar con valores de hasta 65535.

Ambos bancos de registros se corresponden mutuamente, el BC con el B'C', el DE con el D'E' y el HL con H'L'. Todos estos pueden ser usados indistintamente por el programador excepto en algunas instrucciones que requieren registros específicos.

Los registros AF y A'F' son un tanto especiales y de ello habremos de tratar más extensamente en su momento.

El registro A es el registro acumulador por el que pasan todos los resultados operativos de ocho bits y a él están referidas la mayoría de las instrucciones. Es, por tanto, un registro muy específico. El registro F o de FLAGS es el que lleva el estado del microprocesador en todo momento. Cada uno de sus ocho bits tiene un cometido, incluso los dos que en la figura dos aparecen rayados.

EL MICROPROCESADOR Z-80 (III)

Primitivo de FRANCISCO

Tratamos en esta tercera parte sobre el registro de estado del Z-80 (también llamado registro de flags) y el significado de sus bits. Incluimos también una colección de los principales diagramas de tiempos del funcionamiento del Z-80. Estos gráficos han de ser de gran utilidad para los estudiosos de los montajes hardware así como para la comprensión del funcionamiento del Spectrum.

El registro de estado es una serie de ocho biestables, cada uno de ellos almacena temporalmente un bit. El significado de cada uno de estos bits está directamente relacionado con el funcionamiento interno del microprocesador. Su conocimiento es determinante para la comprensión de algunas de las instrucciones del Z-80 y por tanto para su programación.

Cuando se empieza a utilizar el código máquina pronto se ve la potencia funcional que tiene este registro el cual es laborioso acceder y modificar por software; ya que está concebido para que sea el propio microprocesador el que lo altere y, aunque puede ser leído por software, su cometido principal reside en que su estado es consultado por el Z-80 al ejecutar las instrucciones aritméticas, lógicas, etc.

De los ocho bits hay dos (el 3 y el 5) que no tienen significado para el programador, pero sí para el microprocesador, pues se apoya en ellos para la ejecución de ciertas operaciones internas. En principio los hemos excluido de nuestro estudio porque es el propio fabricante del Z-80 quien no suministra información oficial en absoluto sobre ellos. Aunque a base de laboriosos resultados empíricos hemos obtenido ciertos resultados que facilitaremos

cuando los tengamos concluidos. Los seis bits restantes son utilizados por el microprocesador así:

– BIT 0 (C): CARRY (ACARREO). Este bit es puesto a uno por el Z-80 cuando el resultado de operar un dato rebosa en el acumulador el máximo valor posible con ocho bits 255 ó FF en hexadecimal, en caso contrario permanece a cero. Este bit puede ser afectado por las instrucciones aritméticas, lógicas, por las de rotación y desplazamiento. Si se operan datos de 16 bits con los registros dobles, entonces el bit C se pone a uno cuando el resultado de una operación excede a 65535 ó FFFF hexadecimal. Este bit también actúa de borrow o bit de acarreo en instrucciones de substracción.

– BIT 1 (N): ADD/SUBSTRACT (SUMA/RESTA EN BCD). Este bit es actuado en operaciones en BCD. BCD significa: Decimal Codificado en Binario en que obviamente están prohibidos los números hexadecimales A hasta F. Para trabajar en BCD se utilizan las mismas instrucciones que operan en hexadecimal, pero el resultado final requiere un ajuste decimal mediante la instrucción DAA, la cual utiliza la información de los bits 1 y 4 del registro de estado. En la operación de resta en BCD este bit se pone a 1 y a 0 en las de

suma lógica, de rotación y desplazamiento.

– BIT 2 (P/V): PARITY/OVERFLOW (PARIDAD Y DESBORDAMIENTO). Este bit es utilizado como indicador de bit de paridad o como bit de desbordamiento según las instrucciones.

Las instrucciones aritméticas le usan como indicador de desbordamiento. Se produce desbordamiento cuando el resultado de una operación aritmética excede al rango entre -128 (80H) y +127 (7FH). En operaciones de ocho bits. Similarmente existe desbordamiento en operaciones de 16 bits cuando el resultado excede a rango -32768 (8000 H) y +32767 (7FFFH).

Este bit del registro de estado es utilizado como bit de paridad en las instrucciones lógicas, de rotación y desplazamiento. Estas lo ponen a uno si en el resultado la suma de bits que están a 1 es par y a 0 si es impar.

– BIT 4 (H): HALF CARRY (ACARREO MEDIO INTERDIGITOS). Este bit tiene sólo sentido en operaciones en BCD en donde el acarreo que se produce cada cuatro bits es determinante para su operatividad en sumas y restas. Recordemos que la instrucción DAA produce un ajuste decimal del resultado en hexadecimal en una operación aritmética. En BCD únicamente se utilizan las combinaciones binarias desde 0000 hasta 1001 (esto es desde 0 a 9) por tanto las otras seis combinaciones desde 1010 hasta 1111 (desde A hasta F el hexadecimal) producen acarreo entre los dos dígitos BCD que entran en cada byte.

– BIT 6 (Z): ZERO (RESULTADO CERO). Este bit es alterado por las instrucciones aritméticas y lógicas y también por la de comprobación de bit.

En general el bit Z se pone a 1 si el resultado de una operación es 0, y se pone a 0 cuando el resultado es distinto de cero. En este bit también se apoyan las instrucciones que implican decisión como son las de salto, que ejecutan o no la bifurcación según el resultado entregado en este bit producido por la instrucción inmediatamente anterior.

– BIT 7 (S): SIGN (SIGNO DEL RESULTADO). Este bit es alterado por las instrucciones aritméticas, lógi-

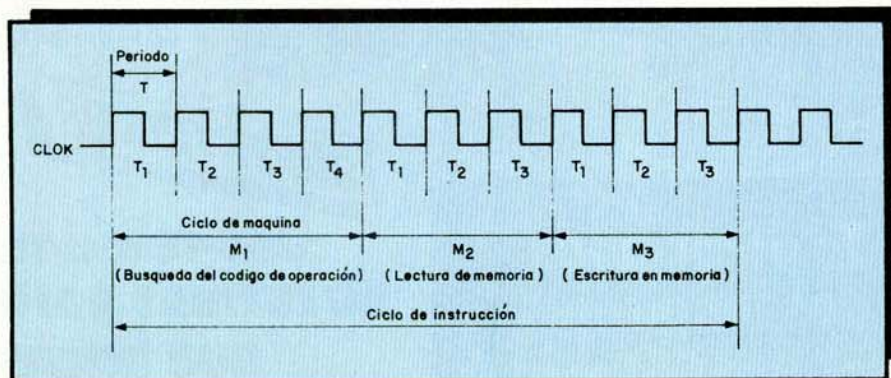


Fig. 1. Impulsos de reloj que precisa el Z-80 para los diferentes momentos clave en el funcionamiento de este microprocesador.

cas, de rotación y desplazamiento. Su significado consiste en indicar el signo del resultado de alguna operación de las indicadas. En el bit S del registro de estado queda copiado el bit de mayor peso de una operación. Así serán positivos los números comprendidos entre 0 y 7FH porque su bit número 7 es un cero, y negativos desde 80H hasta FFH porque su bit número 7 es un uno.

En las operaciones cuyo resultado implica una palabra ó 16 bit, el bit S se pone a uno en resultados comprendidos entre 8000H y FFFH, es decir, cuando el bit de mayor peso (el número 15) es un uno y son positivos entre 0000H y 7FFFH porque el bit 15 es un cero.

El registro de flags va unido al acumulador cuando ambos son utilizados como registro de 16 bits y se encuentra duplicado en ambos bancos de registros. Esto es útil a la hora de guardar temporalmente su contenido para luego recuperarlo tras la ejecución de algún fragmento de programa.

Los diagramas de tiempos

Los diagramas de tiempos, que presentamos seguidamente, contienen toda la información de cómo trabaja el Z-80 en sus diferentes fases a lo largo del tiempo de ejecución de un programa. En ellos se ve cómo evolucionan las distintas señales que hemos comentado en anteriores artículos.

En base a estas se diseñan los distintos periféricos que se conectan al Spectrum, y por supuesto, también nuestros montajes. En la figura número 1, se muestran los tres estados de ejecución por los que discurre constantemente el Z-80 en relación a la señal de reloj. El reloj, como ya se ha dicho, está fijado en el Spectrum a 3,5 Mhz; por tanto, su periodo de instrucción, denominados: M1, M2 y M3, emplea cuatro periodos de reloj, lo que equivale a 1,140 microsegundos. Estos tiempos son los que determinan la velocidad del Z-80, así si suponemos, por ejemplo, que una determinada instrucción requiere tres ciclos, ésta tardará en ejecutarse 3,42 microsegundos. Este tiempo parece minúsculo y despreciable desde nuestra perspectiva de movimientos humanos, pero resulta enorme a nivel electrónico. Esto se comprueba cuando una sucesión de instrucciones va acumulando microsegundos dando de resultas un tiempo notable en la ejecución de una rutina.

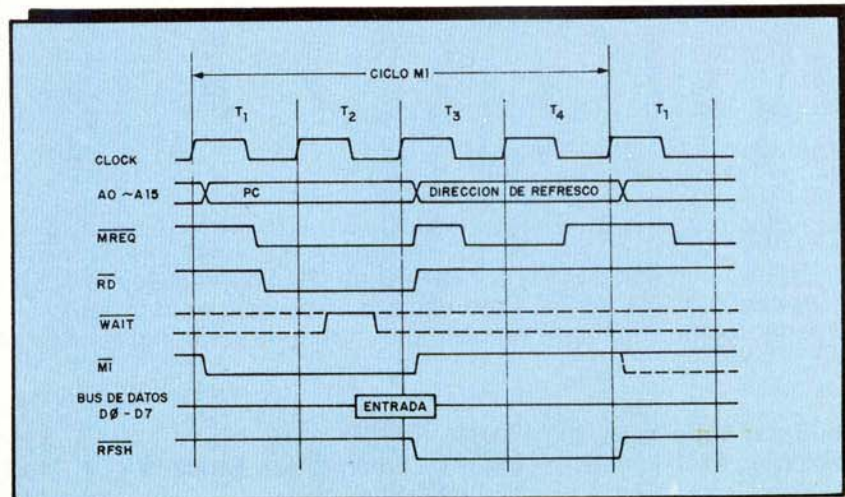


Fig. 2. Diagrama de estados lógicos en el momento de tomar el byte de código de una instrucción. En el dibujo se muestra la evolución de las señales del Z-80 que intervienen en este proceso.

En toda instrucción, el primer ciclo lo emplea el Spectrum para hacer la búsqueda del código de instrucción y decodificarlo, el segundo y tercer ciclo se suelen emplear en el acceso a memoria para la recogida del resto de los bytes que componen la instrucción. En la figura número 2, se representa el estado de las señales en que el Z-80 accede a la memoria para leer un byte. Si éste es el código de una instrucción, se genera durante dos periodos de reloj la señal M1.

pretende la ralentización de los buses para el uso de memorias o periféricos lentos.

En el momento cumbre de toda transparencia se habilita el bus de datos cuyo movimiento es, en esencia, lo que se pretende en cada ciclo de acceso a memoria o puertos.

Las señales RD y WR son determinantes en sus ciclos respectivos de lectura o escritura. Como se observa en las distintas figuras estas señales son

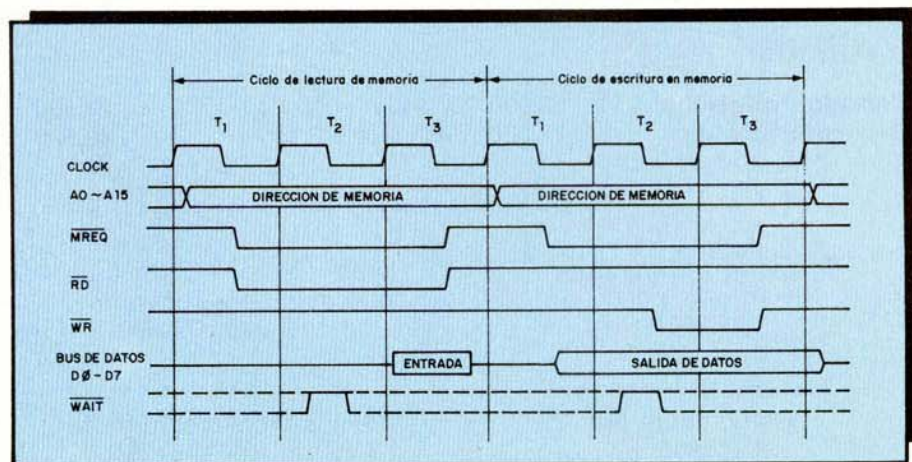


Fig. 3. Diagrama de estados lógicos en el momento de leer o escribir en memoria el Z-80.

En todo momento se han de alternar el ciclo de acceso o memoria y el de refresco de la misma. Obsérvese cómo en el acceso intervienen las señales MREQ y RD si es lectura, WR si es escritura (ver figura número 3).

La señal RFSH se produce durante el tiempo de refresco de las memorias RAM dinámicas. La señal WAIT que aparece en los diagramas es opcional y sólo se genera desde el exterior si se

activas en su estado bajo como la mayoría de las que intervienen en el Z-80. El estado alto es de reposo. En la figura número 5 se muestra el estado de las señales para el acceso a puertos tanto para escribir como para leer. Es obvio que RD y WR no pueden darse al tiempo aunque así represente en la figura, en donde se han comprendido sinópticamente ambos ciclos.

EL MICROPROCESADOR Z-80 (IV)

Primitivo de FRANCISCO

En esta cuarta parte continuamos con los diagramas de tiempo, con el resto de los registros internos y su utilidad para el hardware. Finalmente analizaremos la filosofía del conexionado entre el Z-80 y el resto de la circuitería que le es imprescindible para funcionar.

El Z-80 tiene una capacidad de direccionamiento de 65536 puertos, idéntica a la de acceso a memoria. La semana pasada decíamos que es la señal MREQ la que orienta la dirección hacia el área de memoria, que podría alcanzar también hasta 65.536 bytes o direcciones posibles (esto viene determinado por los 16 bits de dirección del Z-80). Entendemos por memoria al conjunto de celdas en que es posible ubicar datos y programa.

El Z-80, mediante la señal IORQ, es capaz de acceder a otro bloque de direccionamiento que se denomina zona de puertos. Esta se caracteriza por estar únicamente reservada al trasiego de datos entre periféricos y microprocesador. En el direccionamiento de puertos también sería posible conectar memoria RAM (únicamente estática); pero el Z-80 sólo accedería para escribir o leer datos, nunca para ejecutar un programa. Sin embargo, en la zona de direccionamiento de memoria sí se podrían conectar periféricos. En el Spectrum de 48K esto resulta complicado ya que tiene ocupadas por ROM y RAM la totalidad de sus direcciones. En el acceso a puertos la señal IORQ es la que lleva todo el peso, es ésta la que conmuta el

resto de los buses hacia los periféricos. Como es lógico, de lo dicho se deduce que MREQ e IORQ son antagónicas y nunca pueden darse al mismo tiempo; pero de esto no ha de preocuparse el usuario, ya que la lógica interna del Z-80 no lo permite. IORQ actúa en conjunción con el resto de las señales, en especial con RD y WR para leer y escribir respectivamente en alguna dirección de puerto. En el diagrama de tiempo de la Figura 1, se puede observar la evolución de las principales señales que intervienen en el acceso a puertos. En el dibujo se han representado ambos tiempos, el de lectura y el de escritura para economizar espacio; pero obviamente nunca se podrán generar al mismo tiempo. Hay que recordar que todas las señales de control son activas únicamente al pasar a estado bajo, que es lo que quiere indicar la rayita que se coloca encima del nombre de la señal.

En la Figura también se ha incluido la señal *wait* la cual sirve para detener el progreso del resto de las señales en tanto que ella está a nivel bajo. Nótese en la ilustración cómo se pierde un ciclo de reloj (TW) como consecuencia de haber actuado WAIT en algún momento anterior próximo en la línea de

puntos.

La señal *wait* tiene importancia especial en el acceso a puertos, ya que éstos pueden ser muy variados en cuanto a velocidad de conmutación y *wait* puede permitir una perfecta sincronización adaptando en todo momento la velocidad de la CPU al periférico. Únicamente cabe recordar que al llevar a nivel bajo la señal *wait* cesan los ciclos de refresco de las RAM dinámicas, por tanto ésta nunca deberá sobrepasar los 2 milisegundos en estado bajo. La utilidad de esta señal la veremos pronto en los montajes que tenemos en preparación.

Los registros de indexado y misceláneos

Existen en el Z-80 dos registros de 16 bits cada uno cuyo fin es el de servir de puntero para operaciones de indexado. El indexado es usado por el software para el acceso a tablas de datos ubicadas en cualquier lugar de la memoria. Los dos registros son denominados IX e IY, en principio ambos pueden ser utilizados indistintamente, aunque si en el programa se accede a subrutinas de la ROM es mejor dejar intacto el registro IY que apunta constantemente a las direcciones de las variables del sistema.

Estos dos registros requieren un byte que será añadido a la dirección apuntada por el registro, a éste se le llama byte de desplazamiento. La dirección absoluta formada por el registro más el desplazamiento servirá de índice a la instrucción que usa IX o IY. El desplazamiento puede ser hacia adelante o hacia atrás según el bit número 7 del citado byte. El puntero de stack (SP) es otro puntero cuya base material es un registro de 16 bits. SP está apuntando constantemente a una dirección de memoria que se fija por software. Esta posición forma parte del stack en donde se almacenan automáticamente las direcciones de retorno de subrutina o los datos manipulados mediante las instrucciones PUSH o POP, que utilizan el stack como almacén eventual del

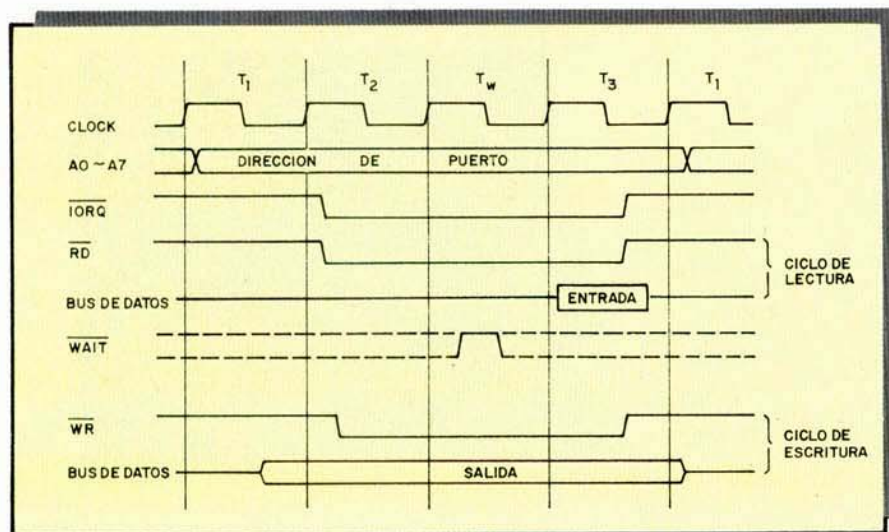


FIGURA 1. Diagrama de los estados lógicos de las señales que toman parte en el Z-80 para dirigirse a puertos en los momentos de escritura o de lectura.

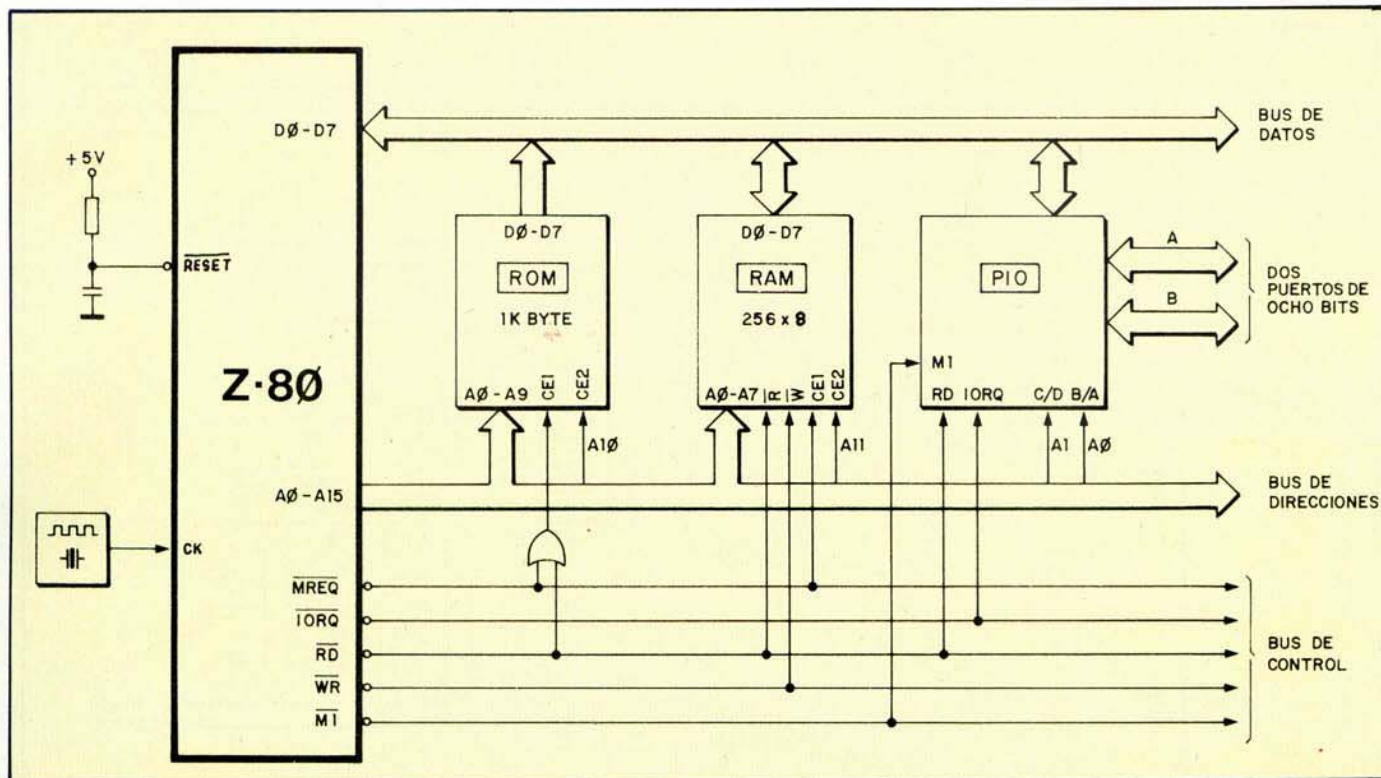


FIGURA 2. Ejemplo de construcción de un sistema mínimo con el Z-80 incluyendo ROM, RAM y una PIO (controlador de periféricos) en aplicaciones generales.

contenido de aquellos registros dobles que han sido procesados por las citadas instrucciones.

El stack es un área de RAM tratada de modo secuencial similar a las memorias LIFO. Estas siglas provienen del inglés: Last In, First Out, esto es, el último dato introducido será el primero en salir al ser leído.

El contador de programa (Program Counter), es un registro de 16 bits que sirve para apuntar a la dirección de memoria en que se está ejecutando el programa. El microprocesador lo incrementa a medida que va accediendo a memoria para tomar los bytes de que se componen las instrucciones. Su contenido puede verse bruscamente alterado al ejecutarse instrucciones de salto relativo o absoluto.

El registro de interrupción (I) es de ocho bits y se usa para conformar la dirección absoluta del vector de interrupción. Las interrupciones desde el punto de vista hardware son algo más complejas y requieren, sin duda, un tratamiento aparte, cosa que haremos en un futuro inmediato con objeto de su utilización en nuestros montajes. Por último, el registro de refresco (R) es de ocho bits y es utilizado como contador de refresco por el propio microprocesador el cual lo incrementa automáticamente después de cada operación de búsqueda de instrucción. Su contenido

sale hacia el exterior por los ocho bits de menor peso del bus de dirección (A0 - A7) para mover las columnas de refresco de la RAM dinámica. Este registro no es normalmente utilizado por el programador, aunque si tiene acceso para fines de prueba.

En la Figura 2 se representa un ejemplo del sistema mínimo para el Z-80. El microprocesador no puede funcionar por sí solo, necesita el auxilio de ciertos dispositivos que se han de conectar a sus buses. Precisa de un reloj para mover toda su lógica interna (en la Figura el reloj está representado por un cuadrado que encierra el símbolo de un cristal de cuarzo).

Precisa también de un circuito inicializador que le produzca un impulso de reset. A sus buses de datos han de ir conectadas una ROM y una RAM, esta última suponemos que es estática para mayor simplicidad. La ROM contendrá el programa de aplicación y ha de ir conectada a partir de la dirección 0000H que es la dirección inicial de ejecución tras un reset. En las direcciones de RAM se colocará el puntero del stack. En general la RAM servirá para las operaciones en que se requiera almacenar temporalmente tablas o datos producidos en los resultados parciales de un programa.

El sistema mínimo que estamos comentando no tendría ninguna utilidad

si no fuera por la PIO que es un circuito integrado capacitado para hacer de interface con el mundo exterior con el cual se ha de relacionar.

Un sistema mínimo sólo tiene sentido en aplicaciones de automatizaciones por microprocesador y en fines didácticos.

El describir aquí un sistema mínimo para el Z-80 nos sirve para ver objetivamente y de manera bastante simplificada cómo se conectan los distintos circuitos a los terminales ya mencionados de este microprocesador. El Spectrum y los circuitos que se proponen en esta sección han de cumplir necesariamente esta filosofía de conexionado.

Los buses de datos y de dirección han de llegar a los distintos bloques de ROM, RAM, PIO, etc. De estos buses se conectarán a cada dispositivo los bits que precise, por ejemplo, la ROM de nuestro dibujo, que hemos supuesto de 1 Kbyte, necesitará los ocho bits del bus de datos y diez bytes del bus de direcciones ($1K=1024=2^{10}$) estos bits se tomarán a partir del menor peso, es decir desde A0 hasta A9. A10 se utiliza como bit decodificador de la ROM que irá conectado a una entrada de chip select. Este tipo de memorias son de «Sólo lectura» por lo que a la otra entrada de selección la habrán de llegar las señales MREQ (indicativa de la zona de memoria) y RD (señal de lectu-

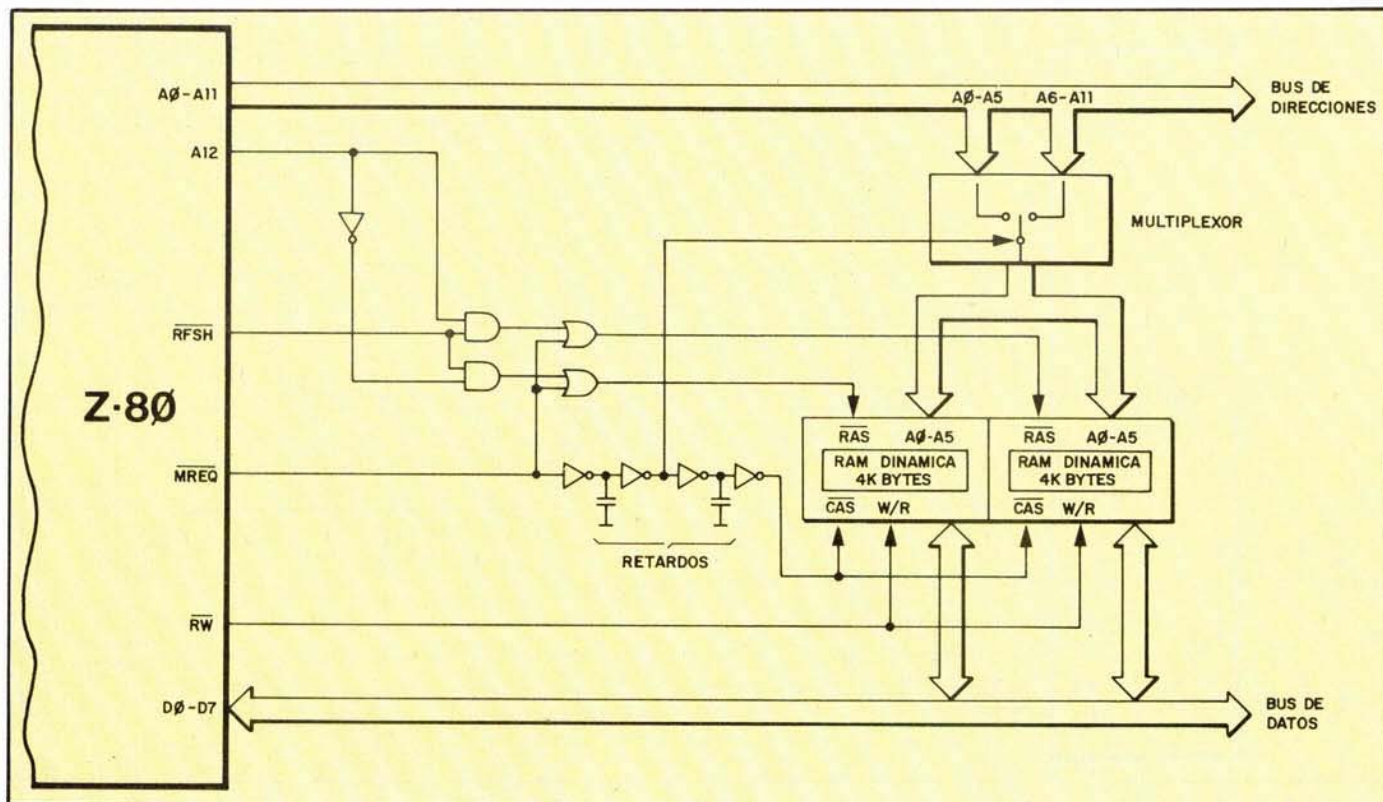


FIGURA 3. Esquema del conexionado de memorias dinámicas al microprocesador Z-80.

ra), ambas se suman mediante una puerta OR. La RAM, que hemos supuesto de 256 bytes, se conecta similarmente a la ROM, sólo que precisa también de la señal WR para ser escrita. El bit A11 del bus de direcciones es el usado para la selección de la RAM.

Todo circuito que entre hacia el bus de datos ha de cumplir los requisitos mencionados siendo imprescindible el uso de líneas triestado de las cuales van provistas las distintas memorias y todos los periféricos.

La PIO, que es un circuito integrado específico para la conexión de periféricos al Z-80, es un dispositivo muy apreciado para este cometido pues

simplifica notablemente la circuitería necesaria. Posiblemente nosotros recurramos a él en algún montaje si su costo y utilidad resultasen ventajosos. Obsérvese como el PIO emplea la señal IORQ como habilitadora, síntoma típico de que este dispositivo es específico para el tratamiento de puertos. También se utilizan A0 y A1 para dirección de las posibilidades internas. El resto de las señales responden a requerimientos específicos de la PIO.

Por último, hay que destacar que todas las conexiones han de ir orientadas hacia una perfecta sincronización para el acceso al bus de datos, el cual opera constantemente bajo la fórmula tries-

tado, lo cual significa que sus bits son usados para el trasiego de datos en ambos sentidos. Esto conlleva a que el bus sólo esté disponible para un determinado dispositivo únicamente durante algunos manosegundos, tal como se aprecia en los diagramas de tiempo de capítulos anteriores.

La Figura 3 muestra el esquema de bloques para la conexión de memorias RAM dinámicas al Z-80, en éste las señales RFSH y MREQ son las encargadas de gestionar la lógica del refresco en donde el multiplexor de direcciones y la célula de retardos conforma la totalidad de la circuitería anexa a toda RAM dinámica.

MICROMANIA. Sólo para adictos

HOBBY PRESS, S.A.

Editamos para gente inquieta.

Una revista con marcha para los que necesitan saber **TODO** sobre ordenadores.



EL MICROPROCESADOR Z-80 (y V)

Primitivo de FRANCISCO

Concluimos con esta quinta parte la serie que hemos dedicado al conocimiento del microprocesador Z-80. Con estos artículos hemos pretendido de modo divulgativo dar a conocer como funciona este microprocesador siempre desde la perspectiva hardware con vistas hacia la conexión de periféricos especialmente.

El SPECTRUM viene equipado con memorias RAM dinámicas, gracias a las prestaciones que tiene el Z-80 para este cometido. Si se hubieran incluido estáticas su consumo sería del orden de cinco veces superior junto con una alta necesidad de disipación de calor y un gran espacio en la tarjeta.

Para conectar RAM dinámicas al Z-80 sólo hay que adosarle unos cuantos componentes. En la figura 3 de la semana pasada se incluía un dibujo en el que de manera sinoptica se veía la filosofía de conexionado. Existe un bloque multiplexor encargado de doblar el bus de direcciones bajo la acción de la señal MREQ que en el ciclo de refresco actúa de clock, al tiempo que, con un cierto retardo activa las columnas de refresco en la organización interna de la memoria.

Cuando la señal RFSH se va a nivel bajo, el contenido del registro interno I pasa los bits bajos del bus de direcciones para direccionar las columnas de refresco. Este registro se incrementa en uno constantemente en cada ciclo de refresco pasando de FF a 00 nuevamente como un estado más.

Las células de retardo incluidas en la línea MREQ sirven para evitar solapamientos de los flancos en las señales de refresco. En el ejemplo de la figura el BIT A12 se usa para direccionar una u otra RAM de 4 Kbytes (obsérvese que la señal RAS en el direccionamiento de acceso sólo es permitida cuando no existe RFSH y se produce MREQ. La señal RAS habilita la dirección a las filas según la organización interna en filas y columnas.

Las instrucciones del Z-80

El Z-80 posee 256 instrucciones que se decodifican únicamente por el primer byte de código. Así, por ejemplo, la instrucción "DJNZ,e" es de dos bytes 10 xx de los cuales 10 es el byte de código y xx es el desplazamiento en el salto relativo que representa.

En la figura número 1 se muestra la tabla con todas las instrucciones de este tipo en sus diferentes variaciones según el registro que interviene en el proceso de cada una de ellas.

La tabla está en forma de doble entrada de manera que el byte de código en hexadecimal se forma mediante la numeración de filas y columnas.

La finalidad de esta tabla es la de mostrar de forma esquemática la estructuración de las instrucciones del Z-80 mediante la cual se rige el decodificador de instrucciones entorno al microprocesador. En la tabla se puede observar la ley de formación de las instrucciones y la secuencia que se siguen con respecto a los registros.

En cada cuadrado de la tabla se muestra: El código de la instrucción que se ha subrayado, y el nemónico o nombre por el que se le conoce y el que ha de usarse en los programas fuente. Sobre el nombre y función de las instrucciones no trataremos porque desde hace algunas semanas se está incluyendo en Microhobby un curso de programación en código máquina en donde se explican todas ellas profundamente. Esta tabla puede ser un útil como complemento para aquéllos que hayan decidido seguir el citado curso de las páginas centrales.

En algunas casillas existen unas letras minúsculas que significan:
e = Byte de desplazamiento

dd = Byte de dato
ddd = Doble byte de dato
addr = Dirección absoluta

Esta tabla fue la que confeccionó el fabricante del Z-80 a la hora de distribuir las instrucciones a los 256 códigos posibles con 8 bits. Pero su intención era la de dar más potencia a este microprocesador, por lo que hubo de ampliar a dos bytes el código de las instrucciones extras. Para ello eligió cuatro bytes de código (CB, DD, ED y FD) que los empleó como ventanas que comunican con otras tantas expansiones.

En la tabla 1 mostramos las instrucciones que se corresponden con el byte CB por el cual empiezan. Todas ellas son de dos bytes, el segundo byte cubre casi otras tantas

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00 NOP	01 LD BC, +ddd	02 LD (BC), A	03 INC BC	04 INC B	05 DEC B	06 LD B, +dd	07 RLCA	08 EX AF, A'F'	09 ADD HL, BC	0A LD A, (BC)	0B DEC BC	0C INC C	0D DEC C	0E LD C, +dd	0F RRCA
1	10 DJNZ, e	11 LD DE, +ddd	12 LD (DE), A	13 INC DE	14 INC D	15 DEC D	16 LD D, +dd	17 RLA	18 JR, #	19 ADD HL, DE	1A LD A, (DE)	1B DEC DE	1C INC E	1D DEC E	1E LD E, +dd	1F RRA
2	20 JR NZ, #	21 LD HL, +ddd	22 LD (addr), HL	23 INC HL	24 INC H	25 DEC H	26 LD H, +dd	27 DAA	28 JR Z, #	29 ADD HL, HL	2A LD HL, (addr)	2B DEC HL	2C INC L	2D DEC L	2E LD L, +dd	2F CPL
3	30 JR LD	31 LD SP, +ddd	32 LD (addr), A	33 SP	34 (HL)	35 (HL)	36 (HL), dd	37 SCF	38 JR C, #	39 ADD HL, SP	3A LD A, (addr)	3B SP	3C INC A	3D DEC A	3E LD A, +dd	3F CCF
4	40 LD B, B	41 LD B, C	42 LD B, D	43 LD B, E	44 LD B, H	45 LD B, L	46 LD B, (HL)	47 B, A	48 C, B	49 C, C	4A C, D	4B C, E	4C C, H	4D C, L	4E C, (HL)	4F C, A
5	50 LD D, B	51 LD D, C	52 LD D, D	53 LD D, E	54 LD D, H	55 LD D, L	56 LD D, (HL)	57 D, A	58 E, B	59 E, C	5A E, D	5B E, E	5C E, H	5D E, L	5E E, (HL)	5F E, A
6	60 LD H, B	61 LD H, C	62 LD H, D	63 LD H, E	64 LD H, H	65 LD H, L	66 LD H, (HL)	67 H, A	68 L, B	69 L, C	6A L, D	6B L, E	6C L, H	6D L, L	6E L, (HL)	6F L, A
7	70 LD (HL), B	71 (HL), C	72 (HL), D	73 (HL), E	74 (HL), H	75 (HL), L	76 HALT	77 (HL), A	78 A, B	79 A, C	7A A, D	7B A, E	7C A, H	7D A, L	7E A, (HL)	7F A, A
8	80 ADD A, B	81 ADD A, C	82 ADD A, D	83 ADD A, E	84 ADD A, H	85 ADD A, L	86 ADD A, (HL)	87 A, A	88 A, A	89 ADC A, C	8A ADC A, D	8B ADC A, E	8C ADC A, H	8D ADC A, L	8E ADC A, (HL)	8F ADC A, A
9	90 SUB B	91 SUB C	92 SUB D	93 SUB E	94 SUB H	95 SUB L	96 SUB (HL)	97 SBC A, B	98 SBC A, C	99 SBC A, D	9A SBC A, E	9B SBC A, H	9C SBC A, L	9D SBC A, (HL)	9E SBC A, A	9F SBC A, A
A	AD AND B	AE AND C	AF AND D	A0 AND E	A1 AND H	A2 AND L	A3 AND (HL)	A4 XOR A	A5 XOR B	A6 XOR C	AA XOR D	AB XOR E	AC XOR H	AD XOR L	AE XOR (HL)	AF XOR A
B	BD OR B	BE OR C	BF OR D	B0 OR E	B1 OR H	B2 OR L	B3 OR (HL)	B4 CP A	B5 CP B	B6 CP C	BA CP D	BB CP E	BC CP H	BD CP L	BE CP (HL)	BF CP A
C	CD RET NZ	C1 POP BC	C2 JP NZ, +addr	C3 JP +addr	C4 CALL NZ, +addr	C5 BC	C6 ADD A, +dd	C7 RST 0000	C8 RET Z	C9 RET	CA JP Z, +addr	CB TABLA 1	CC CALL Z, +addr	CD CALL +addr	CE ADC A, +dd	CF RST 0008
D	DD RET NC	DE POP DE	DF JP NC, +addr	D1 GUT CALL (+dd), A	D2 CALL NC, +addr	D3 PUSH BC	D4 PUSH DE	D5 SUB +dd	D6 RST 0010	D7 RET C	DA EXX	DB JP C, +addr	DC CALL C, +addr	DD CALL TABLA 2	DE SBC A, +dd	DF RST 0018
E	ED RET PD	EE POP HL	EF JP PO, +addr	E1 JP (SP), HL	E2 CALL PO, +addr	E3 PUSH HL	E4 AND +dd	E5 RST 0020	E6 RET PE	E7 JP (HL)	EA JP PE, +addr	EB EX DE, HL	EC CALL PE, +addr	ED CALL TABLA 3	EE XOR A, +dd	EF RST 0026
F	FD RET P	FE POP AF	FF JP P, +addr	F1 DI	F2 CALL R, +addr	F3 PUSH AF	F4 OR +dd	F5 RST 0030	F6 RET M	F7 JP SP, HL	FA JP M, +addr	FB EX M, HL	FC CALL M, +addr	FD CALL TABLA 4	FE CP +dd	FF RST 0038

Figura 1. En esta tabla se muestran los códigos de las instrucciones desde 00 hast FF. Los recuadros marcados corresponden a las Tablas 1, 2, 3 y 4 en que se detallan las instrucciones de expansión.

INSTRUCCIONES CB

CB00	RLC B	CB5A	BIT 3, D	CBAD	RES 5, L
CB01	RLC C	CB5B	BIT 3, E	CBAE	RES 5, (HL)
CB02	RLC D	CB5C	BIT 3, H	CBAF	RES 5, A
CB03	RLC E	CB5D	BIT 3, L	CB80	RES 6, B
CB04	RLC H	CB5E	BIT 3, (HL)	CB81	RES 6, C
CB05	RLC L	CB5F	BIT 3, A	CB82	RES 6, D
CB06	RLC (HL)	CB60	BIT 4, B	CB83	RES 6, E
CB07	RLC A	CB61	BIT 4, C	CB84	RES 6, H
CB08	RLC B	CB62	BIT 4, D	CB85	RES 6, L
CB09	RLC C	CB63	BIT 4, E	CB86	RES 6, (HL)
CB0A	RLC D	CB64	BIT 4, H	CB87	RES 6, A
CB0B	RLC E	CB65	BIT 4, L	CB88	RES 7, B
CB0C	RLC H	CB66	BIT 4, (HL)	CB89	RES 7, C
CB0D	RLC L	CB67	BIT 4, A	CB8A	RES 7, D
CB0E	RLC (HL)	CB68	BIT 5, B	CB8B	RES 7, E
CB0F	RLC A	CB69	BIT 5, C	CB8C	RES 7, H
CB10	RL B	CB6A	BIT 5, D	CB8D	RES 7, L
CB11	RL C	CB6B	BIT 5, E	CB8E	RES 7, (HL)
CB12	RL D	CB6C	BIT 5, H	CB8F	RES 7, A
CB13	RL E	CB6D	BIT 5, L	CB90	SET 0, B
CB14	RL H	CB6E	BIT 5, (HL)	CB91	SET 0, C
CB15	RL L	CB6F	BIT 5, A	CB92	SET 0, D
CB16	RL (HL)	CB70	BIT 6, B	CB93	SET 0, E
CB17	RL A	CB71	BIT 6, C	CB94	SET 0, H
CB18	RL B	CB72	BIT 6, D	CB95	SET 0, L
CB19	RL C	CB73	BIT 6, E	CB96	SET 0, (HL)
CB1A	RL D	CB74	BIT 6, H	CB97	SET 0, A
CB1B	RL E	CB75	BIT 6, L	CB98	SET 1, B
CB1C	RL H	CB76	BIT 6, (HL)	CB99	SET 1, C
CB1D	RL L	CB77	BIT 6, A	CB9A	SET 1, D
CB1E	RL (HL)	CB78	BIT 7, B	CB9B	SET 1, E
CB1F	RL A	CB79	BIT 7, C	CB9C	SET 1, H
CB20	SLA B	CB7A	BIT 7, D	CB9D	SET 1, L
CB21	SLA C	CB7B	BIT 7, E	CB9E	SET 1, (HL)
CB22	SLA D	CB7C	BIT 7, H	CB9F	SET 1, A
CB23	SLA E	CB7D	BIT 7, L	CB00	SET 2, B
CB24	SLA H	CB7E	BIT 7, (HL)	CB01	SET 2, C
CB25	SLA L	CB7F	BIT 7, A	CB02	SET 2, D
CB26	SLA (HL)	CB80	RES 0, B	CB03	SET 2, E
CB27	SLA A	CB81	RES 0, C	CB04	SET 2, H
CB28	SRA B	CB82	RES 0, D	CB05	SET 2, L
CB29	SRA C	CB83	RES 0, E	CB06	SET 2, (HL)
CB2A	SRA D	CB84	RES 0, H	CB07	SET 2, A
CB2B	SRA E	CB85	RES 0, L	CB08	SET 3, B
CB2C	SRA H	CB86	RES 0, (HL)	CB09	SET 3, C
CB2D	SRA L	CB87	RES 0, A	CB0A	SET 3, D
CB2E	SRA (HL)	CB88	RES 1, B	CB0B	SET 3, E
CB2F	SRA A	CB89	RES 1, C	CB0C	SET 3, H
CB30	SRL B	CB8A	RES 1, D	CB0D	SET 3, L
CB31	SRL C	CB8B	RES 1, E	CB0E	SET 3, (HL)
CB32	SRL D	CB8C	RES 1, H	CB0F	SET 3, A
CB33	SRL E	CB8D	RES 1, L	CB10	SET 4, B
CB34	SRL H	CB8E	RES 1, (HL)	CB11	SET 4, C
CB35	SRL L	CB8F	RES 1, A	CB12	SET 4, D
CB36	SRL (HL)	CB90	RES 2, B	CB13	SET 4, E
CB37	SRL A	CB91	RES 2, C	CB14	SET 4, H
CB40	BIT 0, B	CB92	RES 2, D	CB15	SET 4, L
CB41	BIT 0, C	CB93	RES 2, E	CB16	SET 4, (HL)
CB42	BIT 0, D	CB94	RES 2, H	CB17	SET 4, A
CB43	BIT 0, E	CB95	RES 2, L	CB18	SET 5, B
CB44	BIT 0, H	CB96	RES 2, (HL)	CB19	SET 5, C
CB45	BIT 0, L	CB97	RES 2, A	CB1A	SET 5, D
CB46	BIT 0, (HL)	CB98	RES 2, B	CB1B	SET 5, E
CB47	BIT 0, A	CB99	RES 2, C	CB1C	SET 5, H
CB48	BIT 1, B	CB9A	RES 2, D	CB1D	SET 5, L
CB49	BIT 1, C	CB9B	RES 2, E	CB1E	SET 5, (HL)
CB4A	BIT 1, D	CB9C	RES 2, H	CB1F	SET 5, A
CB4B	BIT 1, E	CB9D	RES 2, L	CB20	SET 6, B
CB4C	BIT 1, H	CB9E	RES 2, (HL)	CB21	SET 6, C
CB4D	BIT 1, L	CB9F	RES 2, A	CB22	SET 6, D
CB4E	BIT 1, (HL)	CB9A	RES 3, B	CB23	SET 6, E
CB4F	BIT 1, A	CB9B	RES 3, C	CB24	SET 6, H
CB50	BIT 2, B	CB9C	RES 3, D	CB25	SET 6, L
CB51	BIT 2, C	CB9D	RES 3, E	CB26	SET 6, (HL)
CB52	BIT 2, D	CB9E	RES 3, H	CB27	SET 6, A
CB53	BIT 2, E	CB9F	RES 3, L	CB28	SET 7, B
CB54	BIT 2, H	CB9A	RES 3, (HL)	CB29	SET 7, C
CB55	BIT 2, L	CB9B	RES 3, A	CB2A	SET 7, D
CB56	BIT 2, (HL)	CB9C	RES 3, B	CB2B	SET 7, E
CB57	BIT 2, A	CB9D	RES 3, C	CB2C	SET 7, H
CB58	BIT 3, B	CB9E	RES 3, D	CB2D	SET 7, L
CB59	BIT 3, C	CB9F	RES 3, E	CB2E	SET 7, (HL)
		CBAC	RES 5, H	CB2F	SET 7, A

Tabla 1. Expansión de las instrucciones de 2 bytes que comienzan por CB.

INSTRUCCIONES DD

DD09	ADD IX, BC
DD19	ADD IX, DE
DD21	LD IX, NN
DD22	LD (NN), IX
DD23	INC IX
DD29	ADD IX, IX
DD2A	LD IX, (NN)
DD2B	DEC IX
DD34	INC (IX + d)
DD35	DEC (IX + d)
DD36	LD (IX + d), N
DD39	ADD IX, SP
DD46	LD B, (IX + d)
DD4E	LD C, (IX + d)
DD56	LD D, (IX + d)
DD5E	LD E, (IX + d)
DD66	LD H, (IX + d)
DD6E	LD L, (IX + d)
DD70	LD (IX + d), B
DD71	LD (IX + d), C
DD72	LD (IX + d), D
DD73	LD (IX + d), E
DD74	LD (IX + d), H
DD75	LD (IX + d), L
DD77	LD (IX + d), A
DD7E	LD A, (IX + d)
DD86	ADD A, (IX + d)
DD8E	ADC A, (IX + d)
DD96	SUB (IX + d)
DD9E	SBC A, (IX + d)
DDA6	AND (IX + d)
DDAE	XOR (IX + d)
DDB6	OR (IX + d)
DDBE	CP (IX + d)
DD01	POP IX
DD03	EX (SP), IX
DD05	PUSH IX
DD09	JP (IX)
DD0F	LD SP, IX
DDCBXX06	RLC (IX + d)
DDCBXX0E	RRC (IX + d)
DDCBXX16	RL (IX + d)
DDCBXX1E	RR (IX + d)
DDCBXX26	SLA (IX + d)
DDCBXX2E	SRA (IX + d)
DDCBXX3E	SRL (IX + d)
DDCBXX46	BIT 0, (IX + d)
DDCBXX4E	BIT 1, (IX + d)
DDCBXX56	BIT 2, (IX + d)
DDCBXX5E	BIT 3, (IX + d)
DDCBXX66	BIT 4, (IX + d)
DDCBXX6E	BIT 5, (IX + d)
DDCBXX76	BIT 6, (IX + d)
DDCBXX7E	BIT 7, (IX + d)
DDCBXX86	RES 0, (IX + d)
DDCBXX8E	RES 1, (IX + d)
DDCBXX96	RES 2, (IX + d)
DDCBXX9E	RES 3, (IX + d)
DDCBXXA6	RES 4, (IX + d)
DDCBXXAE	RES 5, (IX + d)
DDCBXXB6	RES 6, (IX + d)
DDCBXXBE	RES 7, (IX + d)
DDCBXXC6	SET 0, (IX + d)
DDCBXXCE	SET 1, (IX + d)
DDCBXXD6	SET 2, (IX + d)
DDCBXXDE	SET 3, (IX + d)
DDCBXXE6	SET 4, (IX + d)
DDCBXXEE	SET 5, (IX + d)
DDCBXXF6	SET 6, (IX + d)
DDCBXXFE	SET 7, (IX + d)

Tabla 2. Expansión de las instrucciones de 2, 3 y 4 bytes que comienzan por DD.

INSTRUCCIONES FD

FD09	ADD IY, BC
FD19	ADD IY, DE
FD21	LD IY, NN
FD22	LD (NN), IY
FD23	INC IY
FD29	ADD IY, IY
FD2A	LD IY, (NN)
FD2B	DEC IY
FD34	INC (IY + d)
FD35	DEC (IY + d)
FD36	LD (IY + d), N
FD39	ADD IY, SP
FD46	LD B, (IY + d)
FD4E	LD C, (IY + d)
FD56	LD D, (IY + d)
FD5E	LD E, (IY + d)
FD66	LD H, (IY + d)
FD6E	LD L, (IY + d)
FD70	LD (IY + d), B
FD71	LD (IY + d), C
FD72	LD (IY + d), D
FD73	LD (IY + d), E
FD74	LD (IY + d), H
FD75	LD (IY + d), L
FD77	LD (IY + d), A
FD7E	LD A, (IY + d)
FD86	ADD A, (IY + d)
FD8E	ADC A, (IY + d)
FD96	SUB (IY + d)
FD9E	SBC A, (IY + d)
FDA6	AND (IY + d)
FDAE	XOR (IY + d)
FDB6	OR (IY + d)
FDBE	CP (IY + d)
FDE1	POP IY
FDE3	EX (SP), IY
FDE5	PUSH IY
FDE9	JP (IY)
FD0F	LD SP, IY
FDCBXX06	RLC (IY + d)
FDCBXX0E	RRC (IY + d)
FDCBXX16	RL (IY + d)
FDCBXX1E	RR (IY + d)
FDCBXX26	SLA (IY + d)
FDCBXX2E	SRA (IY + d)
FDCBXX3E	SRL (IY + d)
FDCBXX46	BIT 0, (IY + d)
FDCBXX4E	BIT 1, (IY + d)
FDCBXX56	BIT 2, (IY + d)
FDCBXX5E	BIT 3, (IY + d)
FDCBXX66	BIT 4, (IY + d)
FDCBXX6E	BIT 5, (IY + d)
FDCBXX76	BIT 6, (IY + d)
FDCBXX7E	BIT 7, (IY + d)
FDCBXX86	RES 0, (IY + d)
FDCBXX8E	RES 1, (IY + d)
FDCBXX96	RES 2, (IY + d)
FDCBXX9E	RES 3, (IY + d)
FDCBXXA6	RES 4, (IY + d)
FDCBXXAE	RES 5, (IY + d)
FDCBXXB6	RES 6, (IY + d)
FDCBXXBE	RES 7, (IY + d)
FDCBXXC6	SET 0, (IY + d)
FDCBXXCE	SET 1, (IY + d)
FDCBXXD6	SET 2, (IY + d)
FDCBXXDE	SET 3, (IY + d)
FDCBXXE6	SET 4, (IY + d)
FDCBXXEE	SET 5, (IY + d)
FDCBXXF6	SET 6, (IY + d)
FDCBXXFE	SET 7, (IY + d)

Tabla 4. Expansión de las instrucciones de 2, 3 y 4 bytes que comienzan por FD.

256 posibilidades (desde 00 hasta FF), aunque entre 2F y 38 se observa una discontinuidad de 8 bytes, así pues las instrucciones CB 30 hasta CB 37 no existen.

En las tablas 2, 3 y 4 se muestran la expansión de las instrucciones que empiezan por DD, ED y FD. Algunas de éstas pueden llegar hasta 4 bytes, éstas son las que tratan los registros IX e IY.

En las tablas el signo X simboliza, un dígito de un dato o un desplazamiento.

Algunas de las instrucciones que tratan los registros IX e IY en las tablas 2 y 4 requieren de hasta tres bytes de código. Esto representa una rara particularidad del decodificador de instrucción que además incluye el byte de desplazamiento entre el segundo y tercer byte de código.

Obsérvese que las expansiones DD, ED y FD no usan todas las combinaciones del byte que le sigue, sino algunas que le han sido propicias al diseñador por ejemplo la instrucción DD 24 no existe.

Este sistema de ampliación de instruc-

ciones es original del Z-80 y es uno de los factores que le proporcionan su gran potencia, aunque hay que reconocer que esto es a costa de una mayor lentitud operativa al tratar estas instrucciones, pues ha de procesar hasta cuatro bytes de código.

INSTRUCCIONES ED

ED57	LD A, I	ED44	NEG	ED67	RRO
ED58	IN E, (C)	ED45	RETN	ED68	IN L, (C)
ED59	OUT (C), E	ED46	IM 0	ED69	OUT (C), L
ED5A	ADC HL, DE	ED47	LD I, A	ED6A	ADC HL, HL
ED5B	LD DE, (NN)	ED48	IN C, (C)	ED6F	RLD
ED5E	IM 2	ED49	OUT (C), C	ED72	SBC HL, SP
ED60	IN H, (C)	ED4A	ADC HL, BC	ED73	LD (NN), SP
ED61	OUT (C), H	ED4B	LD BC, (NN)	ED78	IN A, (C)
ED62	SBC HL, HL	ED4D	RETI	ED79	OUT (C), A
ED40	IN B, (C)	ED50	IN D, (C)	ED7A	ADC HL, SP
ED41	OUT (C), B	ED51	OUT (C), D	ED7B	LD SP, (NN)
ED42	SBC HL, BC	ED52	SBC HL, DE	EDA0	LDI
ED43	LD (NN), BC	ED53	LD (NN), DE	EDA1	CPI
		ED56	IM 1		

Tabla 3. Expansión de las instrucciones de 2 y 4 bytes que comienzan por ED.