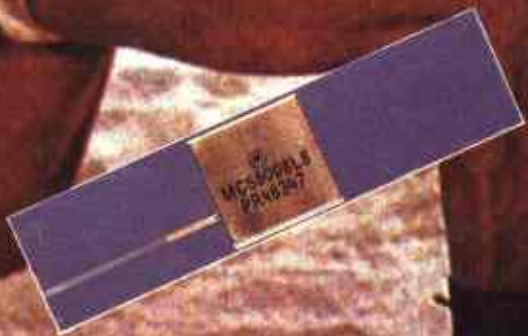




MAGAZINE

Suplemento especial

68008 VERSUS Z-80



Compilación de SuperBASIC



68008

VERSUS

Z-80

Los primeros microprocesadores que se construyeron estaban enfocados a tareas muy determinadas, con un juego de instrucciones reducido. El Intel 8080 puede considerarse el microprocesador capaz de resolver una gama mayor de problemas con el mismo juego de instrucciones.

El 8080 era un microprocesador de 8 bits, y aunque la compañía le retocó para que pudiera trabajar con canales de 16 bits, no había suficiente espacio en memoria para operar con ellos. Debido a estas limitaciones creó el Z-80, al cual se le añadieron 450 instrucciones oficiales, aunque durante el proceso de diseño, surgieron alrededor de una centena de «no oficiales» que no se mencionan en ningún manual, como puede ser el conjunto de instrucciones SLL (*Shift Left Logical*) que brillan por su ausencia en la página 184 del manual del Spectrum (30H a 37H).

El Z-80, aunque sigue siendo un microprocesador de 8 bits, puede operar con 16 bits en el bus de direcciones. (De esta forma es capaz de direccionar 2^{16} (65536) celdas de memoria).

68008 CONTRA Z-80

El **Motorola 68008** es una versión de 8 bits de su antecesor el

Existen muchos usuarios con cierta experiencia en código máquina del microprocesador Z-80, debido sin lugar a dudas al gran auge del Spectrum. Algunos de estos usuarios, al adquirir un QL, se encontraron con un mundo completamente nuevo, ya que el Motorola 68008 poco tiene que vez con el familiar Z-80.

Programar directamente el 68008 no consiste sólo en tener acceso a unos bytes de más y a una potencia operativa mayor, sino una filosofía de programación completamente nueva.

68000 de 32 bits. Mientras el Z-80 fue diseñado para abarcar una gran cantidad de operaciones (es uno de los más completos del mercado), el 68008 (así como el 68000 y toda su familia) está orientado hacia la simplicidad de la programación.

En el Z-80 cada uno de los registros tiene su especial idiosincrasia.

Por ejemplo, para los *ports* de entrada-salida sólo puede usarse el registro C; en bucles rápidos, el B; IX e IY son los únicos que aceptan índices; para operaciones aritméticas el A, y así sucesivamente. Sin embargo, tanto el 68008 como todos los miembros de su familia pueden operar indistintamente con cualquier juego de registros para lo que deseamos hacer.

DISTINTOS REGISTROS

El Z-80 trabaja indistintamente con registros de 8 y 16 bits, poseyendo estos últimos una conformación particular (están constituidos por 2 registros de 8 bits, siendo el 1.º el menos significativo el 2.º el más significativo). Los registros de uso general (de 8 bits) son: B, C, D, E, H y L además del acumulador (registro A), siendo los dobles BC, DE, HL y AF. Aparte de todos éstos, existe un grupo de Registros Alternativos y los registros índices IX e IY así como el I (dirección de página de interrupción) y R (refresco de memoria).

En cuanto al 68008, tiene 8 registros de direcciones (A0 a A7) que contienen las direcciones de memoria que está utilizando el programa, y otros 8 registros de datos (D0 a D7) que calcula y almacenan datos en memoria. Aparte

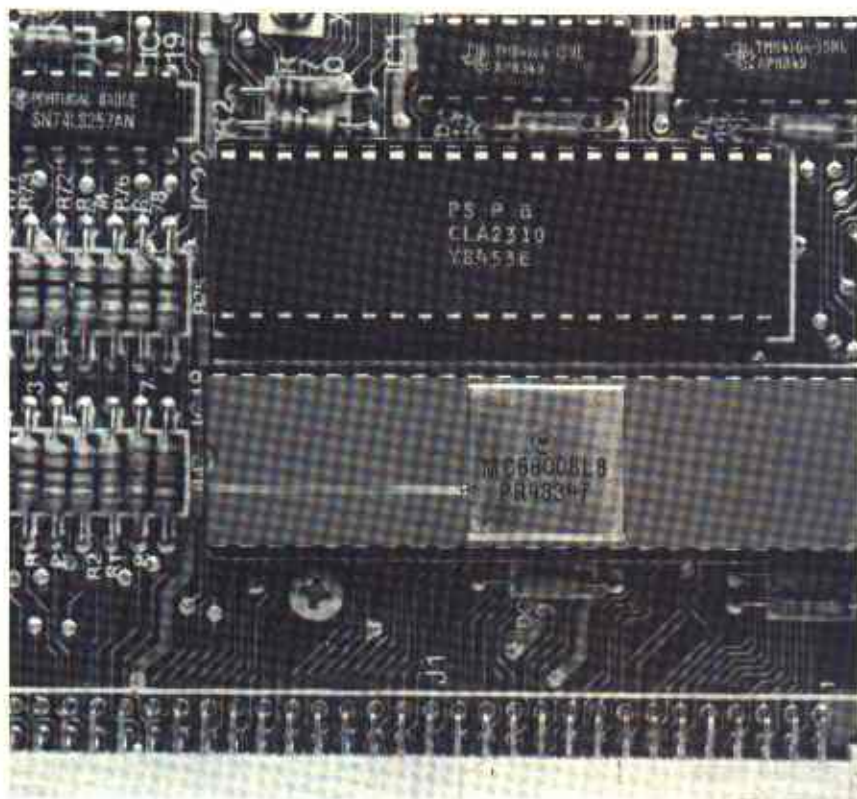


de estos dos juegos de registros, también tiene un contador de programa y un registro de estado. Todos los registros utilizados tienen una longitud de 32 *bits*, menos el registro de estado con sólo 16. Esto hace que la programación en código máquina de 68008 sea muy consistente y facilita el aprendizaje del juego de instrucciones. Aquí ya no es necesario enfrentarse con las sutilezas del Z-80, y la simetría resultante hace más fácil la depuración de errores, alteración de programas y creación de compiladores. En este último sentido es mucho más difícil escribir un buen compilador para el Z-80 debido a la amplia gama de casos especiales que deben ser reconocidos en vistas de producir el mejor código objeto posible. En el extremo, un buen compilador para el 60008 puede producir un código ha partir de un lenguaje de alto nivel como el «C» casi tan eficiente como uno que se hubiese escrito a mano. Esta es la razón por la que el QDOS y el sistema operativo UNIX se han podido escribir enteramente con un compilador en lugar de un ensamblador. (Los programas compilados se escriben más fácilmente que los ensamblados).

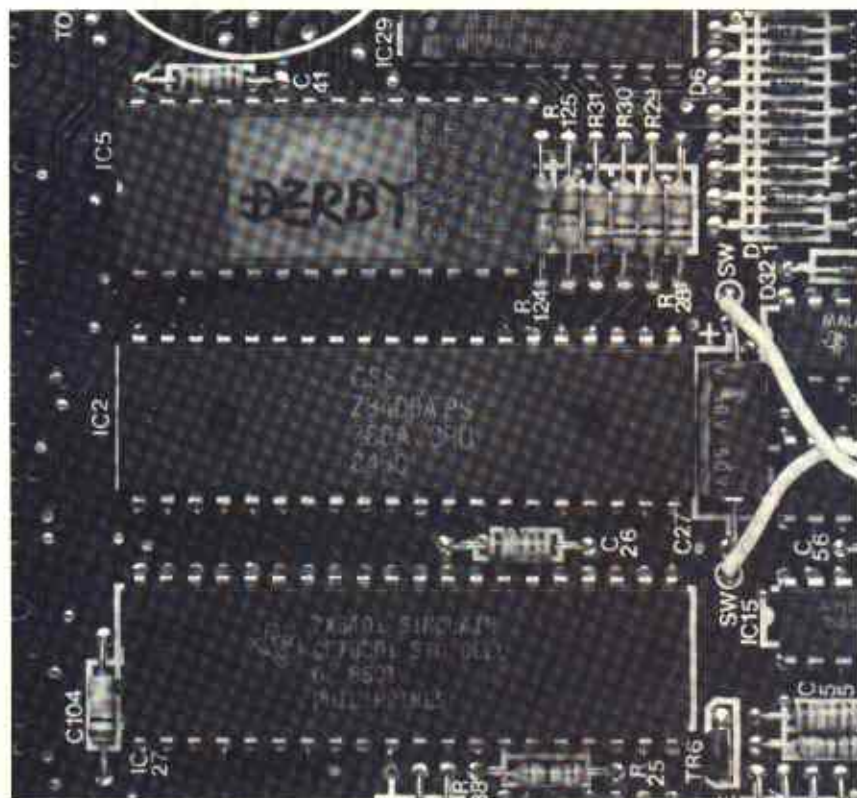
Instrucciones complejas

En teoría, un micro de 16 *bits* es dos veces más potente que uno de 8 *bits*, ya que puede procesar dos veces más información en un mismo paso. Sin embargo, en la práctica, la diferencia es mucho más grande debido a la necesidad de emplear pasos intermedios para convertir dos datos de 8 *bits* en un resultado de 16 *bits*. A diferencia del Z-80, el 68008 tiene la facultad de producir un resultado de 16 de forma inmediata a pesar de que sólo transfiere 8 *bits* a un mismo tiempo cada vez que transmite información.

El 68008 tiene incorporado en su juego de instrucciones algunas muy complejas que hacen en un sólo paso ciertas tareas en las cua-



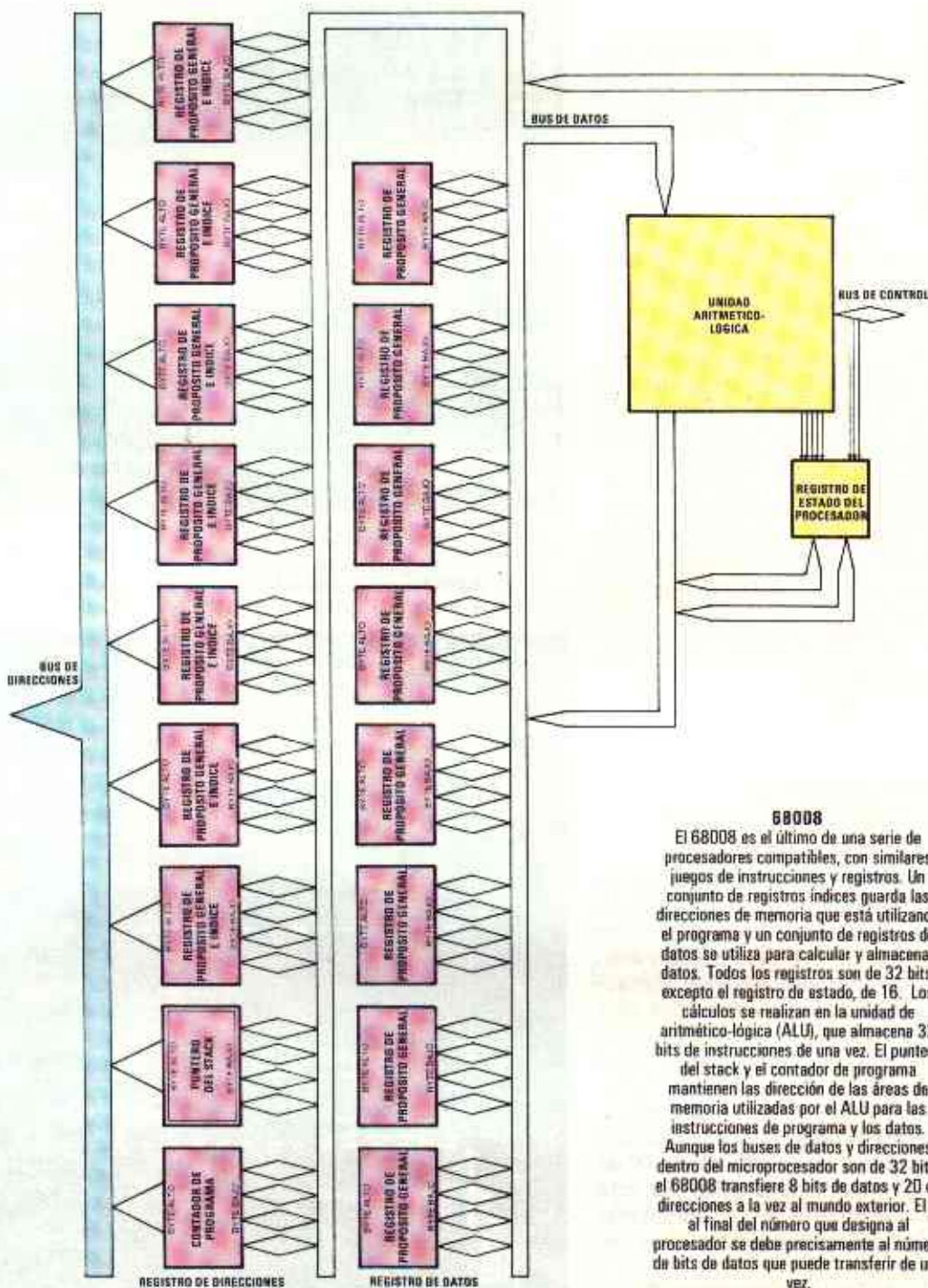
El 68008 se reconoce fácilmente en el QL por la chapa de oro que lo cubre.



Z-80 en el Spectrum 128, junto a la EPROM «Derby».

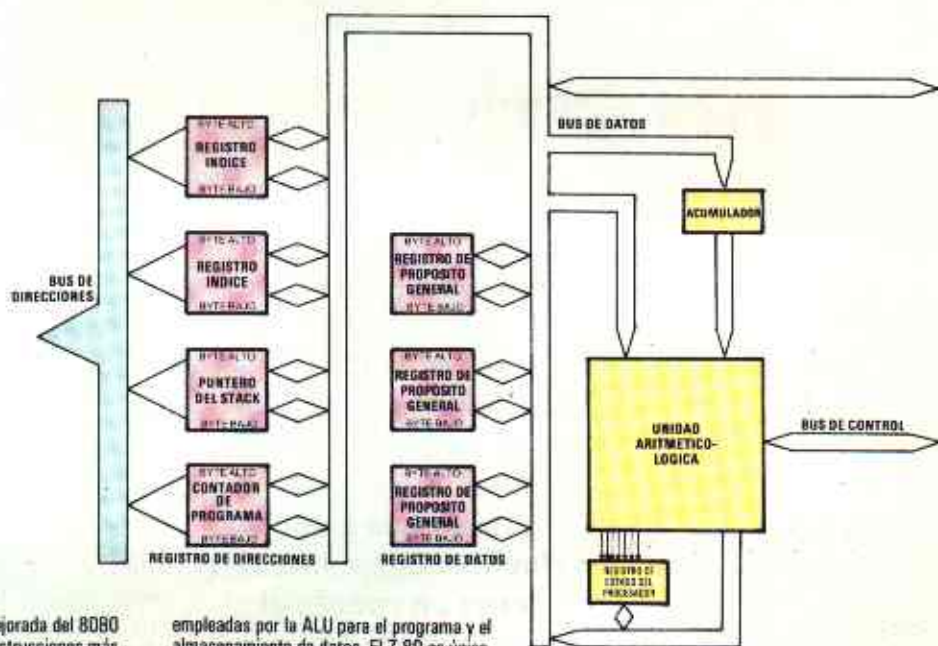
■ El juego de instrucciones del 68008 está orientado hacia la facilidad de programación. ■ El 68008 es capaz de hacer una división o una multiplicación de 32 bits en un solo paso frente al centenar que requiere el Z-80. ■ La simetría del 68008 hace más fácil la depuración de errores, alteración de programas y creación de compiladores.

LA CPU 68008



68008
 El 68008 es el último de una serie de procesadores compatibles, con similares juegos de instrucciones y registros. Un conjunto de registros índices guarda las direcciones de memoria que está utilizando el programa y un conjunto de registros de datos se utiliza para calcular y almacenar datos. Todos los registros son de 32 bits, excepto el registro de estado, de 16. Los cálculos se realizan en la unidad de aritmético-lógica (ALU), que almacena 32 bits de instrucciones de una vez. El puntero del stack y el contador de programa mantienen las direcciones de las áreas de memoria utilizadas por el ALU para las instrucciones de programa y los datos. Aunque los buses de datos y direcciones dentro del microprocesador son de 32 bits, el 68008 transfiere 8 bits de datos y 20 de direcciones a la vez al mundo exterior. El 8 al final del número que designa al procesador se debe precisamente al número de bits de datos que puede transferir de una vez.

LA CPU Z-80



Z-80

El Z-80 es una versión mejorada del 8080 que incluye un juego de instrucciones más versátil y un conjunto de reserva de sus principales registros. Todos los registros son de 16 bits, excepto el acumulador, de 8 bits. (Los otros 8 bits se utilizan para el registro de estado o de banderas). IX e IY se utilizan como registros índice de tablas. El puntero stack y los registros de programa mantienen las direcciones

empleadas por la ALU para el programa y el almacenamiento de datos. El Z-80 es único, pues tiene también un registro de 8 bits de interrupciones y un registro de refresco. El registro de interrupciones contiene los 8 bits superiores de una dirección de interrupción, mientras los ocho inferiores los proporcionan los periféricos. El registro de refresco se encarga del refresco de memoria, eliminando así otro chip externo.

les un microprocesador más sencillo invertiría muchos más. Por ejemplo, el 68008 puede realizar multiplicaciones y divisiones de 32 bits en un único y rapidísimo paso, cuando el Z-80 necesitaría emplear alrededor de un centenar para hacer la misma operación, al no tener implementadas estas funciones.

El 68008 tampoco sacrifica flexibilidad al utilizar sus registros de 32 bits, pues cada instrucción puede emplearse en la forma de 32, 16 y 8 bits.

ACCESO AL C/M DESDE EL BASIC Y EL SUPERBASIC

En el Spectrum, la forma más fácil de ejecutar un programa en código máquina consiste en el uso de la instrucción USR «dirección de comienzo», habiendo antes protegido la zona de memoria donde va almacenado el código mediante CLEAR. Sin embargo, el QL no utiliza la llamada USR, sino CALL seguido de un conjunto de parámetros opcionales. Mediante la instrucción RESPER (X) se reservan X bytes de memoria.

Un programa de QL que utilice código máquina puede tener la siguiente apariencia general:

```
1000 REM RESERVA AREA
1010 LET A = RESPR (100)
1020 PRINT "RESERVADOS
100 BYTES DESDE"; A
1030 FOR X = 1 TO 100
1040 READ D: POKE A + X, D
1050 NEXT X
1060 CALL A, D0, D1, D2, D3,
D4, D5, D6, D7, A0, A1, A2, A3,
A4, A5
```

```
1070 REM VUELVE SI D0 = 0
1080 DATA 1, 2, etc.
```

Finalmente, otra gran diferencia entre el 68008 y el Z-80 radica en el modo de leer el primero los datos almacenados en memoria. El 68008 emplea la técnica llamada *pipelining*. El Z-80 (y cualquier otro microprocesador que no lleve incorporada dicha técnica) trabaja realizando las siguientes tres acciones una por una y en el orden descrito: búsqueda de instrucción, decodificación y ejecución de la instrucción. Sin embargo, el 68008 usa una unidad de separada para leer instrucciones de tal manera que pueda seguir leyendo las siguientes instrucciones mientras el resto del microprocesador está ejecutando la última. Este sistema permite aumentar de manera eficiente la velocidad de proceso.

Orlando Araujo Martín



Compilación de SuperBASIC

Antes del lanzamiento del QL a la calle, se decía que el SuperBASIC incorporado en él, iba a ser un lenguaje potente, multitarea y que trabajaría a una velocidad constante sin tener en cuenta el tamaño del programa. Después vino la gran desilusión: el SuperBASIC no era multitarea, resultaba más bien lento y sólo tenía 7 dígitos de precisión. Para remediar esto, en julio de 1984 Digital Precision empezó a trabajar en el SuperCharge, un compilador automático destinado a convertir el SuperBASIC en rapidísimo código máquina...

Digital Precision pretendía resolver todos los problemas, sin afectar para nada la potencia del lenguaje.

Supercharge debería compilar la inmensa mayoría de los programas existentes en SuperBASIC, sin ninguna alteración. También tendría que ser capaz de funcionar en todas las versiones de QL existentes, con un consumo mínimo de memoria, y el código objeto generado debería ejecutarse a la máxima velocidad posible. De esta manera, se llevó a buen término el diseño del SuperCharge.

El SuperBASIC es lento porque tiene que buscar cada comando, línea y variable de un programa donde quiera que estén. Es un lenguaje interpretado y por lo tanto cada punto y coma, cada instrucción, son verificados detenidamente mientras se ejecuta el programa.

EL ANALIZADOR DEL LENGUAJE

El compilador SuperCharge está formado por dos partes principales: el *parser*, que analiza las instrucciones del programa original en SuperBASIC, y un generador de código objeto, que convierte las instrucciones a las correspon-

dientes en código máquina. Le acompañan también un programa de demostración y un conjunto de procedimientos para el control de tareas y depuración de errores.

El *parser* fue inicialmente escrito en 2.500 líneas de SuperBASIC sin un sólo GO TO o GOSUB. ¡Su primer gran desafío fue compilarse a sí mismo!

El *parser* lee el programa que va a ser compilado y el resultado de sus análisis se almacena en una zona libre de la RAM del QL. Esto produce una rápida compilación y deja libres 40 K de memoria (en el QL estándar) para que el programa que se va a compilar pueda caber perfectamente.

EL GENERADOR DE CÓDIGO

El generador de código es un programa de poca longitud que se carga sobre el *parser*. Está escrito en código máquina para que pueda tratar *bits* y *bytes* a la máxima velocidad.

Este generador código selecciona las rutinas en código máquina que realizarán la compilación. El código producido se graba en un *diskette* o *microdrive*, de tal modo que pueda ser cargado con EXEC. Los programas compilados se car-



El compilador de Digital Precisión está protegido con Leuslock.

gan rápidamente, ya que no tienen que ser «tokenizados» por el SuperBASIC. Debido a estos programas no pueden listarse, están a salvo de ojos indiscretos y de los intentos de piratería.

VENTAJAS Y LIMITACIONES DE SUPERCHARGE

Existen algunas diferencias entre programas compilados e interpretados además de la velocidad.

En el *SuperCharge*, la primera diferencia que se observa es la representación numérica con 9 dígitos de precisión.

SuperCharge se adopta automáticamente a diferentes ROMs y juegos de comandos (transcurren unos momentos antes de la ejecución mientras busca las rutinas necesarias).

El compilador solamente prohíbe unas pocas instrucciones (como ED y RENUM) que cuentan con la presencia de las estructuras de datos del intérprete. Si tuviese que estar al corriente de la fuente del programa y la lista del nombre, sería apenas un poco más veloz que el intérprete.

Otra limitación consiste en que una matriz no debe de contener más de 32.767 elementos. Cadenas y números enteros siempre deben estar señalizados con un signo *dólar* o un tanto por ciento. De la

misma manera, los parámetros de GO TO, GO SUB y DATA tienen que ser constantes (no expresiones).

Estas escasas limitaciones son necesarias para que pueda resultar más compacto y rápido. Si se comete algún error, el compilador muestra un mensaje en inglés que la informa y le muestra el lugar exacto del programa donde se produjo el error.

Los programas con *SuperCharge* funcionan a una velocidad nueve veces más rápida que los programas SuperBASIC convencionales. Esta velocidad puede incluso ser cuadruplicada si se excluye en el programa la coma flotante.

Orlando Araujo Martin