



Una obra maestra

Su gran capacidad de direccionamiento hace que el microprocesador 68000 pueda competir en potencia hasta con los ordenadores centrales

El microprocesador 68000 tiene su origen en la serie 6800 de microordenadores de Motorola, tan bien acogida en el mercado, cuyo punto final fue el microprocesador 6809. Motorola reanudó este cabo con su serie 68000 de alto rendimiento, que ha sido posible gracias a los adelantos en técnicas de VLSI (*very large scale integration*: integración a muy grande escala), que permiten albergar más de 100 000 elementos electrónicos lógicos (tales como puertas AND/OR, flip-flops, etc.) en un solo chip de 64 patillas y de apenas 5 cm de longitud. Con esto se consigue una capacidad de direccionamiento muy similar a la del miniordenador DEC PDP-11, con todas las ventajas y la potencia del juego de instrucciones de un ordenador central: todo en un solo chip.

Estudiaremos más adelante esta capacidad de direccionamiento y de instrucciones. De momento, nos vamos a ceñir a la arquitectura global del 68000 para disponer de un modelo útil donde aplicar nuestros programas. La idea de un modelo es importante porque necesitamos limitar el campo de nuestros conocimientos del microordenador a aquellos aspectos que harán que nuestros programas funcionen. Por ejemplo, no necesitamos comprender cómo trabajan los elementos flip-flop, para poder sumar dos números y establecer el resultado en un registro. Pero sí que nos es necesario saber detalles sobre la longitud de un registro y las instrucciones ADD si queremos resolver ese problema. Nuestro modelo queda, por ello, limitado a las funciones imprescindibles para programar un micro.

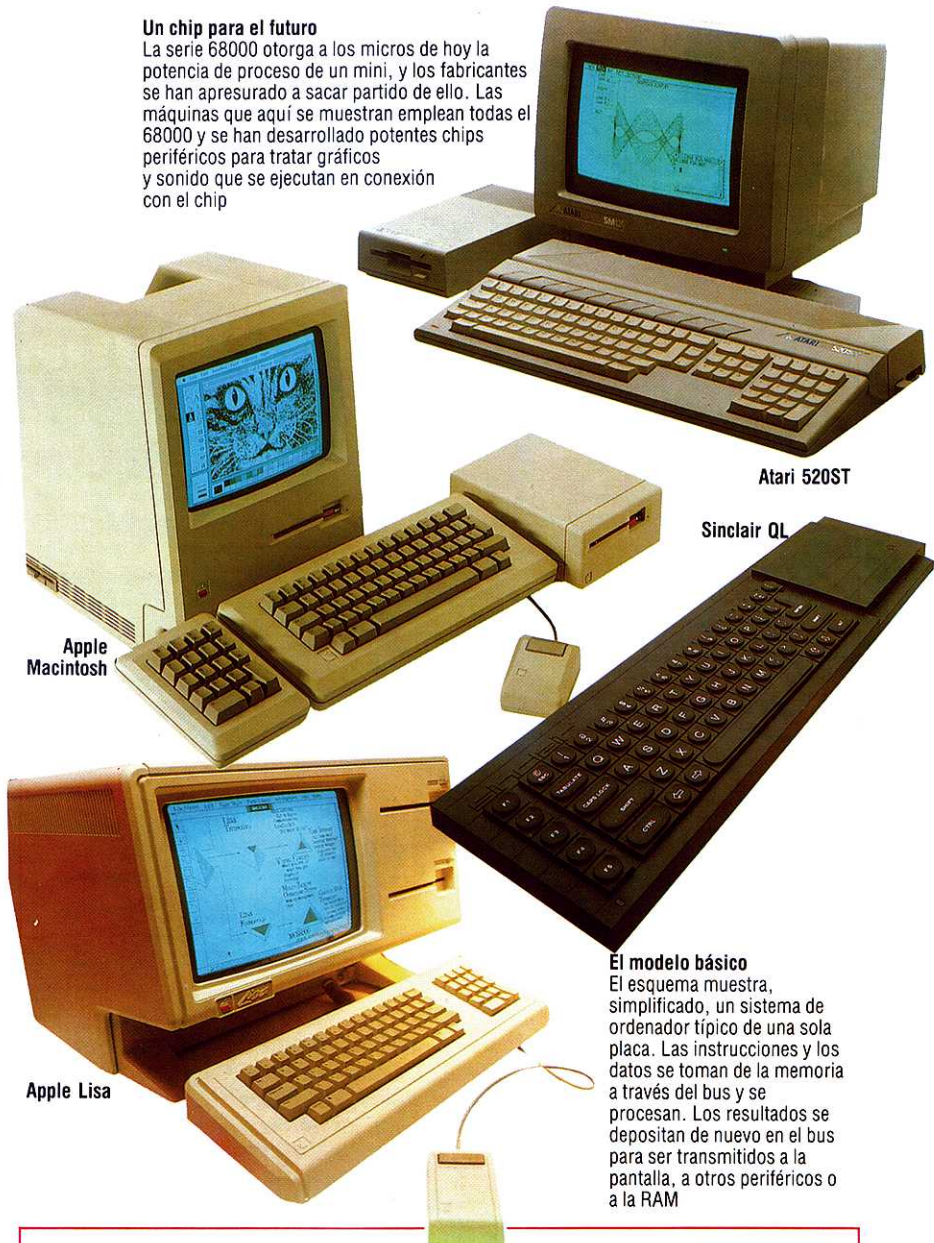
La arquitectura del hardware

Si examinamos las funciones eléctricas de las patillas del microprocesador 68000 veremos que las señales están relacionadas con el control de los buses digitales. Estos buses son los medios de que se vale el microordenador para comunicarse con el mundo electrónico de su entorno y con los cuales se puede construir un sistema de microordenadores. Sobre una única placa es posible construir un típico sistema como el que se esquematiza en el dibujo. Aquí el procesador puede tomar la instrucción siguiente desde la memoria a través del bus, ejecutar la instrucción internamente dentro del procesador y, si fuera necesario, enviar un carácter a un chip de interface para visualizarlo ante el usuario.

Esta arquitectura del hardware constituye la base de nuestro modelo. Todavía necesitamos pensar

Un chip para el futuro

La serie 68000 otorga a los micros de hoy la potencia de proceso de un mini, y los fabricantes se han apresurado a sacar partido de ello. Las máquinas que aquí se muestran emplean todas el 68000 y se han desarrollado potentes chips periféricos para tratar gráficos y sonido que se ejecutan en conexión con el chip



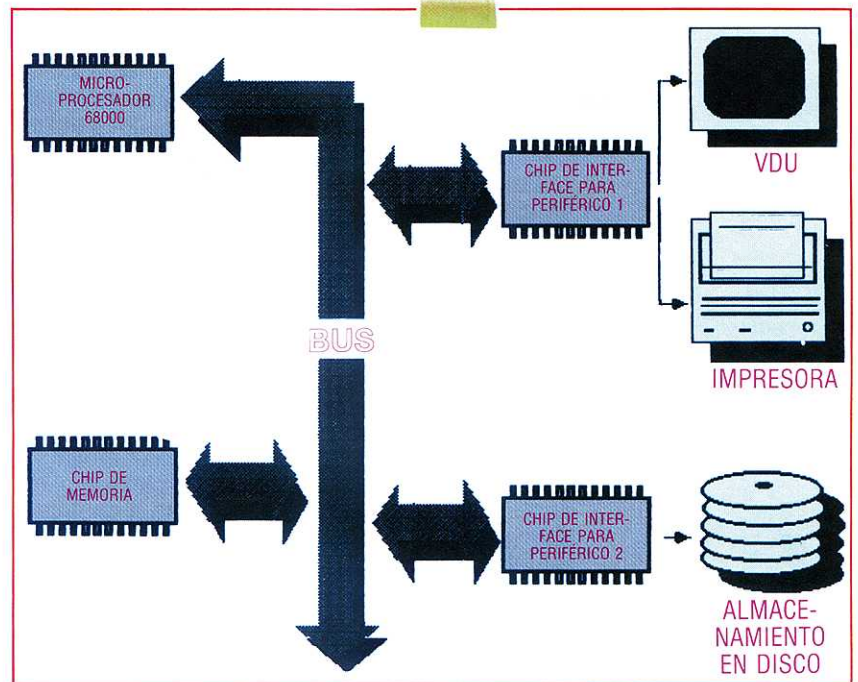
Atari 520ST

Sinclair QL

Apple Macintosh

Apple Lisa

El modelo básico
El esquema muestra, simplificado, un sistema de ordenador típico de una sola placa. Las instrucciones y los datos se toman de la memoria a través del bus y se procesan. Los resultados se depositan de nuevo en el bus para ser transmitidos a la pantalla, a otros periféricos o a la RAM



Mike Clowes

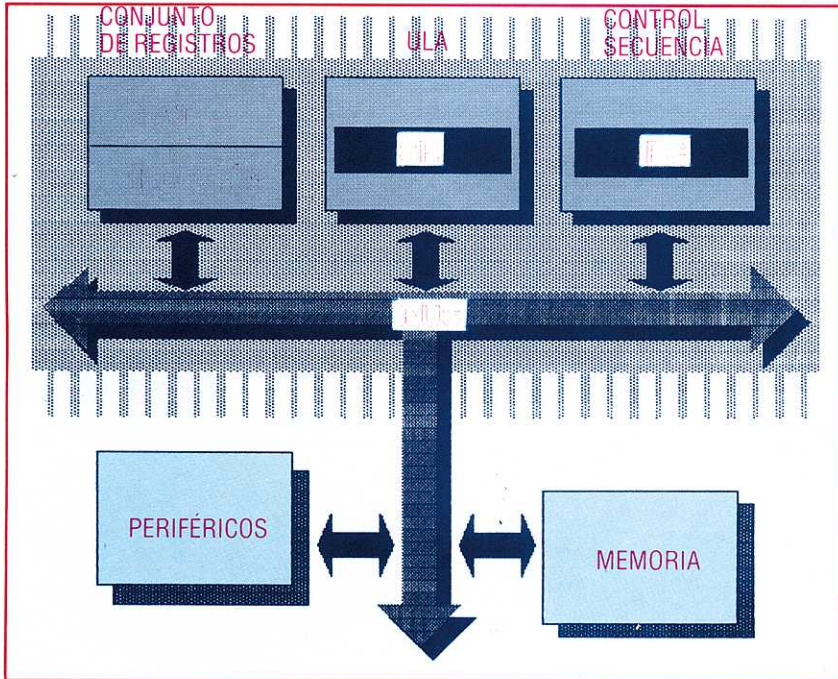


Modelo del programador

El diseño interno del 68000 puede reducirse esencialmente a tres unidades principales: los registros de datos y direcciones, la unidad aritmético-lógica y la unidad de control de secuencia. Cada unidad comunica con las restantes y con los dispositivos periféricos a través del bus de datos digital

como si el microprocesador sólo contara con un bus de comunicación central, pero debemos tener en cuenta también los componentes de nivel de registro interno.

El segundo dibujo muestra el modelo del programador donde podemos ver que la única diferencia en la arquitectura del hardware reside en que el chip del 68000 se ha escindido en tres componentes lógicos. Analicémoslos uno a uno.



Estos registros ya no son, claro está, más que una matriz de rápidas posiciones de memoria empleadas de la misma manera que cualquiera de los registros de otros ordenadores para el almacenamiento de datos o resultados intermedios. La diferencia que caracteriza al 68000 es que cuenta con ocho registros de datos y ocho registros de direcciones formados por 32 bits (dicho de otra manera, son registros de 32 bits de ancho).

No importan las razones de Motorola para escindir los registros en dos subconjuntos, lo cierto es que se han simplificado las operaciones que debe efectuar la máquina y su programación se ha hecho más fácil.

Por ejemplo, si deseamos cargar el contenido de la memoria cuya dirección se encuentra en un registro, listo para una operación aritmética, sabemos con esta arquitectura que el puntero, o la dirección, debe ser cargado en un registro de dirección y que los datos deben ser almacenados en un registro de datos. Traducido al lenguaje assembler, sería así:

```
MOVE (A2),D4
```

donde MOVE es la instrucción de trasladar (o copiar) los datos, A2 significa la dirección de origen como contenido del registro 2 de direcciones, y D4 significa que el destino es el registro 4 de datos. Pero si la dirección de los datos de origen estuviera en D4, no podríamos usar

```
MOVE (D4),A2
```

dado que con MOVE tanto los modos de direccionamiento usados en origen y en destino son ilegales.

Direcciones registradas



Los quince registros del 68000 parece que ofrecen casi un incomparable panorama. Sin embargo, es obligatoria una cierta disciplina a causa de la subdivisión de los registros internos en categorías de direcciones y categorías de datos

Otro detalle a tener en cuenta cuando se consideran los conjuntos de registros, y en particular los registros de datos, es que los datos a los que se debe acceder pueden ser de cinco tipos:

- Bit – un solo dígito binario
- Dígito BCD (o cuarteto)
- Byte – carácter de ocho bits
- Palabra – palabra de 16 bits
- Palabra larga – palabra de 32 bits.

Trataremos los dígitos BCD y las palabras largas en capítulos venideros; de momento los consideraremos como unidades cada vez más grandes de datos.

Gran parte de las instrucciones nos permiten especificar el tipo de dato junto con la instrucción, especificándolo como un atributo de la propia instrucción. Por ejemplo, si sólo deseáramos copiar los bits del 0 al 7 desde D1 a D2 nos serviríamos de:

```
MOVE.B D1,D2
```

pero si deseáramos los 32 bits enteros, la instrucción se transformaría en:

```
MOVE.L D1,D2
```

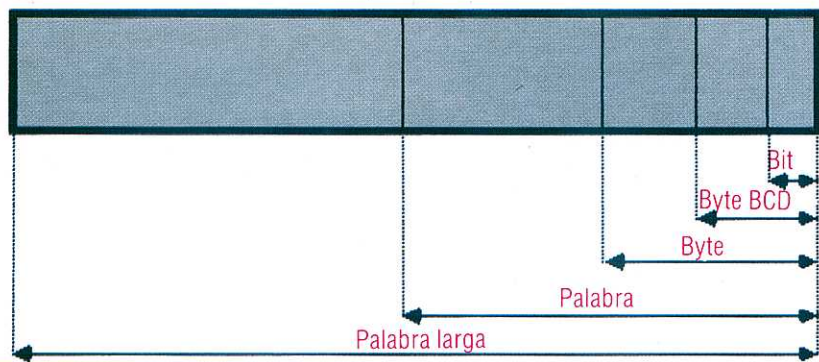
El atributo (o código de tamaño de los datos) para una palabra de 26 bits es .W, que es el atributo por defecto cuando no se especifica ninguno.

Un último punto a notar antes de abandonar el conjunto de registros es que uno de los registros de direcciones, el A7, se emplea como puntero de pila del sistema, por lo que debemos tener cuidado de no cambiar alegremente este registro. Más tarde examinaremos las pilas.



Longitud de las palabras

El 68000 puede, mediante instrucciones ampliadas, operar con palabras de cinco longitudes distintas, procesando datos en grupos cuyo tamaño va de 1 a 32 bits. Lo cual implica las ventajas de la flexibilidad y de una mayor potencia de proceso respecto de los viejos chips de ocho bits



El siguiente elemento por considerar dentro del modelo del programador es la unidad aritmética lógica (ULA). Es la parte del micro encargada de realizar tareas aritméticas o lógicas dejando el resultado en el operando de destino.

Por ejemplo, es la ULA quien suma D1 y D2 y coloca el resultado en D2 cuando se ejecuta la siguiente instrucción:

```
ADD    D1,D2
```

Repitamos que es posible especificar el atributo longitud de los datos con la instrucción:

```
ADD.B  D1,D2
```

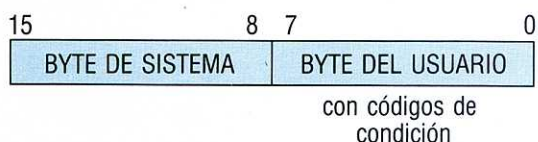
se limitará a sumar los bytes menos significativos.

Volviendo al diagrama del modelo del programador, existe un registro con etiqueta SR, o registro de estado, asociado a la ULA. Este registro está presente de tal modo que podemos recordar el resultado del paso de instrucción anterior. Esto se utiliza para poder efectuar una bifurcación condicional en el programa, según sea el resultado de la ejecución de la instrucción previa. Por ejemplo:

```
ADD    D1,D2    suma D1 a D2
BVS   OFLOW    comprueba desbordamiento
MOVE  D1,D3    copia D1 sobre D3
```

donde una de dos, o bien bifurcamos hasta la etiqueta OFLOW si el bit de desbordamiento está activado en SR, o bien se efectúa MOVE D1, D2 si no lo está. La instrucción BVS (*Branch if overflow Set: bifurcar si desbordamiento activado*) comprueba el desbordamiento o el bit V en el registro de estado, que puede ponerse a uno como resultado de la instrucción ADD (como ocurre en este ejemplo concreto). La condición de desbordamiento surge, como es evidente, a causa de que el resultado de una operación aritmética no se ajusta al tamaño de la palabra empleada en el operando. Si no llegáramos a detectar esta condición obtendríamos respuestas erróneas.

Una última observación sobre el registro de estado: su longitud es de 16 bits, y partes de cada byte son empleadas por el sistema de la siguiente manera:



Los códigos de condición son el conjunto de bits individuales para indicar el resultado de la ejecu-

ción de la instrucción anterior: el bit V es un ejemplo de uno de estos códigos, que se estudiarán profusamente más adelante.

La sección de control de secuencias de nuestro modelo de microprocesador contiene el contador del programa, el registro que determina la dirección de la siguiente instrucción que ha de ser tomada de la memoria. Una vez tomada la instrucción, es decodificada por el control de secuencia para determinar el tipo de instrucción que la ULA ha de ejecutar y dónde están los operandos de origen y destino.

El contador del programa tiene 32 bits de largo pero las patillas a través de las cuales se conecta con el bus sólo permite el uso de 24 bits. Aun así, esto permite una gama amplísima de direcciones de memoria que llega hasta FFFFFFF bytes. Cada dígito hexa corresponde a cuatro bits binarios, por lo que los 24 bits corresponden a un rango de direcciones de byte de 16 777 216. No obstante, se ha de notar que todas las instrucciones deben comenzar en una dirección impar (o frontera de palabra), lo que significa que será más comprensible pensar que hay un máximo de 8 388 608 palabras en ese rango de direcciones.

Ahora que estamos hablando de acceso a la memoria, será conveniente considerar cómo se organizan los datos en ella. Esto es necesario porque los bytes individuales son direccionables en esta máquina, y el direccionamiento de bytes en esta máquina es diferente, por ejemplo, del direccionamiento del PDP-11. El siguiente esquema ilustra el direccionamiento de la memoria del 68000 mostrando cómo la dirección impar proporciona el byte más significativo de la palabra.

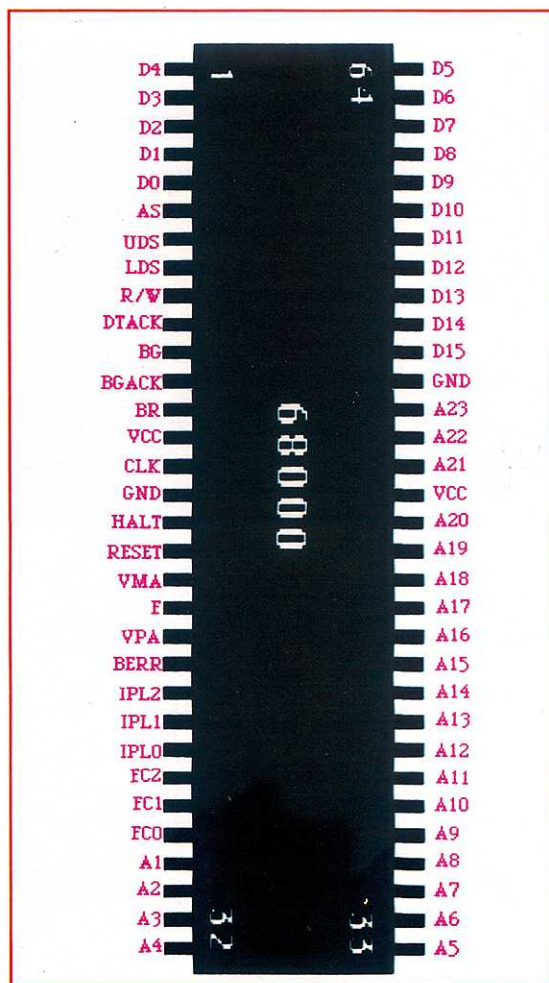
	15	8	7	0
Pal. n	byte n		byte n+1	
Pal. n+2	byte n+2		byte n+3	

Algunas instrucciones cambiarán directamente el contador del programa a fin de provocar una bifurcación, bien sea incondicional como BRA BACKHERE (donde BRA quiere decir *branch always* [bifurcar siempre] a una dirección representada simbólicamente por la etiqueta BACKHERE), bien condicional como en el ejemplo anterior, BVS OFLOW, en el que la bifurcación depende de si el bit V está a uno en la palabra del PS. En ambos casos, cuando ocurre una bifurcación o cambio en el flujo secuencial de ejecución normal, el contador del programa se modifica para apuntar a la siguiente instrucción a ejecutar.



Patillas a granel

Aquí están las patillas de comunicación del 68000. Obsérvense las 24 líneas de direcciones y las 15 de datos. A diferencia de su pariente más próximo, el 68008 (que posee un bus de direcciones de sólo 8 bits), el 68000 puede direccionar en modo directo dentro del ámbito del 000000 al FFFFFFFF hexa, es decir, del 0 al 16 177 216. Su primo mayor, el 68020, tiene un bus de direcciones de 32 bits completo. El 68000 es un perfecto ejemplo de microprocesador moderno: más de 100 000 circuitos lógicos se combinan para obtener la potencia de proceso de un miniordenador en un único chip



El conjunto de instrucciones

Los fabricantes de ordenadores suelen jactarse del número de instrucciones independientes que pueden soportar sus máquinas. Esto puede confundir a más de uno puesto que a menudo es necesaria una instrucción distinta para cada tipo de datos de operando o modo de direccionamiento. Esto es del todo verdadero en los antiguos ordenadores de ocho bits en los que, por ejemplo con el 8085, hay disponibles 63 instrucciones distintas de MOV, y cada una se sirve de los registros en una combinación diferente.

Más importante que el número absoluto de códigos de instrucción es la gama de instrucciones, sus objetos de datos y la flexibilidad de los modos de direccionamiento que pueden ser empleados con las instrucciones.

Motorola ha previsto un conjunto de instrucciones poderosísimo en lo que respecta a objetos de datos, que nos permitirá direccionar bytes, palabras y palabras largas con la mayoría de las instrucciones. Podemos realizar la multiplicación, división binaria y la aritmética BCD, así como la gama habitual de instrucciones de operaciones lógicas, control de programa y copia de datos.

No obstante, aun cuando se dispone de una gama muy útil de modos de direccionamiento, no podemos usarlos sin estudiar antes la instrucción. Por ejemplo, si el operando de destino para la instrucción MOVE es un registro de dirección, debemos emplear la instrucción MOVEA o bien LEA. En otras

palabras, no podemos escribir MOVE D3,A6 pero sí que podemos MOVEA D3,A6. La razón es que al proporcionar un conjunto de instrucciones tan poderoso con respecto a los objetos de datos, se ha sacrificado una pequeña parte de la capacidad de direccionamiento.

En su totalidad el conjunto de instrucciones es muy potente en comparación con los micros de ocho bits (incluidos algunos miniordenadores y ordenadores centrales) con un excelente abanico de objetos de datos, pero con una flexibilidad decepcionante en los modos de direccionamiento utilizables con las instrucciones. En algo se compensa esto gracias a la densidad de algunas instrucciones (conocidas como instrucciones *rápidas*), donde la instrucción entera se codifica en una palabra de máquina.

Por ejemplo, para establecer a 25 un registro de datos escribiríamos:

```
MOVEQ#25,D3
```

que llevará 25 a D3 y la instrucción entera sólo ocupará una palabra, incluyendo el dato constante 25.

En el próximo capítulo estudiaremos con mayor detalle los modos de direccionamiento disponibles y cómo los podemos usar para acceder a los datos. Incluiremos algunos ejemplos de programas.

Por delante de su época

El Motorola 6809 llegó demasiado tarde como para causar impacto en el *boom* de los micros domésticos a finales de los años setenta y comienzos de los ochenta. El 68000 de 16 bits fue lanzado cuando muchos fabricantes estaban diseñando una nueva generación de máquinas, por lo que se ha incorporado a ordenadores como el QL de Sinclair, el Macintosh de Apple y el ST de Atari. Desde el punto de vista del programador, es una delicia trabajar con este chip. El procesador tiene 17 registros de 32 bits, un bus de datos de 16 bits y un bus de direcciones de 24 bits, además del juego de instrucciones.

No obstante, y a pesar del impresionante acabado de muchos de los ordenadores que hasta hoy han incorporado el 68000, existen todavía problemas por resolver antes de que se logre utilizar su completo potencial. El problema básico hasta el momento ha sido el que el 68000 es tan adelantado en su tecnología respecto a otros chips que los fabricantes de ordenadores lo están empleando muy por debajo de sus posibilidades. Un ejemplo extremo es el QL de Sinclair. Este ordenador está basado en la versión 68008 del chip, que sólo tiene un bus de datos de ocho bits. Con lo cual, el QL no es, para muchas aplicaciones, mucho más rápido que sus predecesores de ocho bits. Sin embargo, el futuro del procesador está en el chip 68010, que presenta un bus de direcciones de 16 bits pero que ofrece el soporte de una memoria virtual. Esto permite que parte de la memoria resida en un dispositivo de almacenamiento de fondo más barato, permitiendo así el empleo de mucha más memoria de la que se podría disponer de otro modo. Otro chip de Motorola —el 68020— ha sido saludado como un ordenador de 32 bits en un chip que ofrece el 97 % de la potencia de un ordenador central!