

Discos - Disquetes y discos duros

Entre los dispositivos periféricos existe una gran diferencia: mientras que para acceder a la pantalla y el teclado se utilizan con muy poca frecuencia las funciones del DOS, y se prefieren utilizar las del BIOS, en disquetes y discos duros, esto es justamente al revés. Así como el BIOS sirve ante todo para el formateo físico del disquete o del disco duro, también sirve para el acceso individual a sectores independientes. Pero en la programación DOS esto no es necesario. Ya que normalmente se trata de la manipulación de archivos, que sin el DOS como control central de gestión, no es posible hacer.

Las únicas excepciones son las utilidades de disco, representadas por ejemplo por las PC Tools y las Norton Utilities.

A pesar de ello, se demuestra, entre otros, el acceso a un disquete con ayuda de las funciones del BIOS. Sin embargo no se describe la programación directa de la controladora de disquetes y discos duros, y esto es así por una buena razón: ya que el acceso directo a estos controladores, normalmente sólo genera problemas con controladores incompatibles, y no aporta ninguna ventaja en cuanto a programación. En realidad, prácticamente todo lo que es posible mediante la programación directa, también se puede realizar a través del BIOS.

Y tampoco se consigue ninguna ventaja de velocidad mediante la programación directa, ya que de todas formas se necesita la mayoría del tiempo para el acceso al disco, y el Overhead (gasto de tiempo) de las funciones del BIOS comparativamente no tiene demasiada importancia. Por ello se aconseja que abandone la programación directa de la controladora de disquetes o discos duros.

Aparte de las funciones del BIOS, en este capítulo también se tratan algunos otros temas, que están en relación con el disco duro. Así por ejemplo aprenderá cómo se graban los datos en el disco duro y que diferencias hay entre las diferentes controladoras, que se pueden encontrar en el ámbito del PC.

Estructura de disquetes y discos duros

Lo que une los disquetes y discos duros, es su estructura, lo que naturalmente también se ha condensado en las diferentes funciones del BIOS, que sirven para el acceso a las unidades de disquete o disco duro. Por eso echemos un vistazo a la estructura fundamental de disquetes y discos duros. Comencemos con los disquetes, que en cuanto a su estructura se pueden considerar como discos duros reducidos (de dos dimensiones).

Estructura de un disquete

Los disquetes se dividen fundamentalmente en pistas individuales, que se distribuyen como círculos concéntricos equidistantes sobre la superficie del disco magnético en el interior del disquete. Estas pistas se designan con el número 0 hasta N, donde N es el número total de sectores menos 1, y varía de formato a formato. La pista exterior por norma lleva el número 0, la siguiente el número 1, y siguiendo este esquema se procede hasta la pista más interna.

Cada pista acoge un número constante de sectores, que dividen a esta pista en tramos del mismo tamaño. La cantidad de estos sectores depende de nuevo del formato del disquete y de la unidad. A diferencia de las diferentes pistas, los sectores no se cuentan desde 0, sino desde 1 hasta N, donde N es la cantidad de sectores por pista.

Cada sector contiene 512 Byte y representa la unidad de acceso más pequeña con la que puede operar un programa. Por lo tanto no es posible el leer o escribir un único byte de un disquete, sino que como mínimo se ha de leer o escribir un sector completo.

Las formaciones de bits en uno de estos sectores se registran según el procedimiento llamado FM o MFM, que también se utiliza en discos duros, y que se describe posteriormente. Como programador, no debe pensar en ello durante el acceso a una unidad de disquetes, ya que sólo se llega a tratar con las informaciones binarias que se leen de o se escriben a un sector.

Según su distribución en pistas y sectores, la capacidad de un disquete se calcula según la siguiente fórmula:

$$\text{Pistas} * \text{Sectores por pista} * 512 \text{ Byte por sector.}$$

El valor resultante de esto sólo describe la capacidad de una de las caras del disquete, y se ha de duplicar, si la unidad de disquetes está equipada con dos cabezales de lectura/escritura, ya que en ese caso se utilizan tanto la cara superior como la inferior del disco para guardar datos. Para poder separar las dos caras durante la programación, se denominan con los números 0 y 1.

La cantidad de sectores por pista también determina la denominada tasa de transferencia de datos, que es una medida de la velocidad de trabajo de la electrónica de la unidad de disco y de la controladora conectada. Es importante conocer que con una velocidad de giro constante de 300 revoluciones por minuto, pasan tantos más bits por unidad de tiempo por delante del cabezal de lectura/escritura, cuantos más sectores haya por pista.

Estructura de discos duros

La tasa de transferencia de datos en los discos duros es al menos diez veces mayor que en los disquetes, ya que giran con una velocidad diez veces mayor. Otra potencia de diez se obtiene de nuevo con la tasa de transferencia de datos, ya que los discos duros modernos tienen unos 100 sectores por pista, y eso ya con 3,5 pulgadas de diámetro.

Sin embargo, eso no cambia nada en la estructura fundamental del disco duro, que da la impresión de ser como varios disquetes superpuestos. Ya que en una unidad de disco duro simplemente hay varios discos magnéticos uno encima de otro, que están divididos de la misma forma que los disquetes, en pistas y sectores. Cada disco dispone de dos cabezales de lectura/escritura, que trabajan en la cara superior e inferior del disco.

Mediante un brazo de lectura/escritura están unidos a los demás cabezales, de modo que un cambio de pista se realiza simultáneamente en todos los discos. Esta es la razón, por la cual se han de recorrer todos los cabezales uno tras otro al guardar datos en el disco duro, en vez de cambiar de pista a pista, hasta que la pista correspondiente esté completamente llena. Los cambios de cabezal son mucho más rápidos que el cambio de pista, ya que no están unidos con un movimiento mecánico.

Para describir este movimiento de cabezal a cabezal, se ha introducido el término cilindro. Un cilindro comprende todas las pistas que tienen el mismo número de pista, pero que se encuentran en diferentes discos. Y en realidad ésta es toda la diferencia con respecto a la estructura de un disquete.

Unidades de disquete y formatos

Quien quiera acceder a disquetes con ayuda del BIOS, ha de familiarizarse a diferencia de la programación DOS con los diferentes formatos de disquetes, que son comunes en el ámbito del PC. En el centro de atención se encuentran las diferencias entre unidades de 5 1/4" y 3 1/2", así como la diferenciación entre disquetes de Single- Double- y High-Density.

Disquetes y unidades de 5 1/4"

Al principio de la evolución, estuvieron las unidades de 8", que ya estaban anticuadas, cuando el PC entró en el mundo del proceso de datos. En él se utilizaron directamente las nuevas unidades de 5 1/4", como aún hoy se emplean como equipamiento estándar de la mayoría de PC. Se trata de unidades, que soportan los llamados disquetes de «Double Density». Esta denominación se eligió, para delimitarlos de los disquetes de «Single Density», que anteriormente habían sido utilizados en muchos microordenadores.

Los disquetes «Single Density» simplemente contenían cuatro sectores por pista y 40 pistas por cara, lo que daba una capacidad de 80 KB por cara o 160 KB en unidades con dos cabezales de lectura/escritura. Con 125 KBit/s la tasa de transferencia, comparando con los estándares de las unidades actuales, era bastante baja.

Los disquetes de «Double Density» doblan con respecto a los de Single Density en número de sectores a ocho, donde el número de pistas por cara permanecía igual. Con ello se duplica la capacidad del disquete a 160 KB en los de una sola cara, y a 320 KB en los de unidades de doble cara. Lo mismo sirve para la tasa de transferencia de datos, que en los disquetes Double Density es de 250 KBit/s.

Pero no se permaneció mucho tiempo en este formato, ya que con 8 sectores no se llena del todo una pista en los disquetes de Double Density, sino que queda espacio para un noveno sector. Y este es el responsable del formato tan conocido de 360 KB, con el que trabaja DOS.

Después de los disquetes de Single- y Double-Density, con la introducción del AT apareció el conocido formato de High-Density, que permite la conocida capacidad de 1,2 MB en unidades de 5 1/4. Para ello, no sólo se subió el número de sectores por pista de 8 a 15, sino que también se incrementó el número de pistas hasta 80. Igual que en los disquetes de Double Density, se emplearon las dos caras del disquete.

En teoría se hubieran podido alcanzar 16 sectores, ya que las unidades de los AT trabajan con una tasa de transferencia de datos nuevamente duplicada, de 500 KBit/s. Pero se temió forzar demasiado el material magnético, que hubiera tenido que alcanzar una doble densidad, y por ello se subieron las revoluciones del motor de 300 a 360 revoluciones. Esto vuelve a bajar el número de sectores por pista de 16 a 15.

Denominación	Unidad	Sectores/pista	Pistas/cara	Capacidad	Tasa transferencia
Double Density	PC/XT	8	40	160/320 KB	250 KBit/s
Double Density	PC/XT	9	40	180/360 KB	250 KBit/s
High Density	AT	15	80	1,2 MB	500 KBit/s

Puesto que las pistas individuales del formato High-Density están mas cerca unas de otras (80 en vez de 40 en la misma anchura), este formato no se puede procesar por las antiguas unidades de los PC y XT, sino sólo por las nuevas unidades MF de los AT. MF es abreviación de «multifunción», ya que estas unidades también pueden conmutar al antiguo formato Double Density, y procesar con ello los disquetes normales de 360 KB. Aparte de la tasa de transferencia, también se puede reducir la velocidad de rotación a la medida normal de 300 revoluciones por minuto.

Por otra parte, el mayor número de sectores por pista, impide el procesamiento de disquetes High-Density con las antiguas unidades de los PC y XT. Además, la tasa de transferencia de 500 KBit/s no puede ser alcanzada por estas unidades, ya que simplemente no están diseñadas para ello, y no son configurables (por software).

El aumento de la capacidad de almacenamiento de los disquetes Double-Density a los de High-Density no es sólo una cuestión de electrónica de la unidad, sino sobre todo depende del «granulado» del material magnético en el disquete. Ya que cuanto más pequeñas las diferentes partículas del material magnético, tanto más informaciones se pueden guardar en una superficie determinada. Esta es la razón, por la cual los disquetes Double-Density nunca se pueden formatear sin errores con 1,2 MB en las unidades de los AT, ya que su granulado simplemente es demasiado basto.

Normalmente se consigue formatear una gran parte del disquete, de forma que se consigue una capacidad de unos 600 KB o más, pero a menudo, estos disquetes sólo se pueden leer en el ordenador que los formateó. Otros PC suelen responder con errores de lectura, que hacen el uso de estos disquetes prácticamente imposible. La culpa la tiene un fenómeno que también se ha de tener en cuenta en el correcto formateo de disquetes: el diferente posicionamiento de los cabezales de lectura/escritura en las diferentes unidades de disquete.

A pesar de que sólo es de fracciones de milímetros, se ha de neutralizar, haciendo las diferentes pistas más anchas que el cabezal de lectura/escritura, y posicionando éste siempre en el medio de estas pistas. Contando con una pequeña tolerancia se podrán leer correctamente las informaciones.

Pero esto no funciona en el «sobreformateo» de un disquete Double-Density, ya que la posición del cabezal de lectura/escritura de la unidad formateadora decide sobre si se alcanza una pista por casualidad y si está libre de errores o no. Si la posición del otro cabezal de lectura/escritura se diferencia un poco de la de la unidad original, no tiene esta suerte, y entonces falla por poco en el acierto de una pista.

Un fenómeno parecido se ha de considerar, cuando se formatean disquetes High-Density en unidades Double-Density. Podrá ser que fallen en otras unidades, o al menos generar errores de lectura. Quien cree que le hace un favor a su unidad de disquetes de PC o XT con los caros disquetes de HD (HD: High Density) se equivoca.

Unidades y disquetes de 3 1/2"

A pesar de que las unidades de disquete de 5 1/4" aún están muy extendidas, se sustituyen más y más por las pequeñas unidades de 3 1/2", cuya entrada estelar comenzó con los Laptops, y hoy en día se encuentran en la mayoría de PC de sobremesa o torre. A favor de estos disquetes habla su menor tamaño, pero también su mayor estabilidad, que le deben a su sólida carcasa de plástico. Además, la

ranura de acceso ya no está libre, sino que se encuentra protegida por un deslizador metálico, lo que hace innecesario el guardarlos en una funda de papel.

También en los disquetes y las unidades de 3 1/2" se distingue entre Double Density y High Density. Double Density corresponde a una capacidad de 720 KB, donde se guardan 9 sectores en una pista y donde cada cara del disquete (se emplean las dos) contiene 80 pistas. Así que se obtiene una densidad de pista bastante más alta que en los disquetes de 5 1/4", ya que el diámetro de este disquete es inferior.

Las primeras unidades de 3 1/2", que se emplearon en el ámbito del PC, simplemente soportaban este formato, y por ello quedaban limitadas a una capacidad de 720 KB. La doble capacidad de los disquetes de 1,44 MB la ponen a disposición las unidades y los disquetes de High Density, que hoy se han convertido en el estándar en las unidades de 3 1/2". A pesar de que sólo tienen 80 pistas en cada cara del disquete, formatean cada pista con 18 sectores. Con ello naturalmente duplican la tasa de transferencia de datos, de modo que estos disquetes no se pueden emplear en las unidades Double Density de 3 1/2" de los PC y XT antiguos.

Las unidades de 3 1/2" que pueden procesar el formato de High Density, análogamente a las unidades MF de 5 1/4" están en disposición de adaptarse a disquetes Double Density, de leerlos, escribirlos y formatearlos.

Que la tendencia en disquetes va en dirección 3 1/2" lo demuestran los nuevos disquetes de alta capacidad, que tienen 2,88 MB y que se denominan como Extra-High-Density (ED). Duplican la cantidad de sectores con respecto a los disquetes High Density de nuevo a 36, y por ello sólo se pueden leer en unidades ED. Naturalmente, estas unidades son como siempre, compatibles hacia abajo, de modo que pueden procesar todos los formatos de disquete anteriores.

La unidades que aceptan diferentes formatos de disquete, antes de un acceso han de averiguar el formato del disquete insertado, y pasar esta información al BIOS. En los disquetes de 5 1/4" esto no carece de problemas, ya que esta información sólo se puede obtener con un acceso de lectura, lo que presupone el formateo previo del disquete. En los disquetes de 3 1/2" esto no es así, ya que revelan su capacidad mediante un pequeño agujero, que se encuentra a la altura de la protección de escritura en el lado opuesto de la carcasa de plástico.

Igual que en la protección contra escritura, la existencia de este agujero se puede comprobar con facilidad con un diodo luminoso y un sensor fotoeléctrico. Según las normas, en los disquetes de doble densidad falta este agujero, mientras que en los disquetes de alta densidad este agujero ya se encuentra troquelado en la carcasa. Esto también es cierto para los disquetes Extra-Density, pero en ellos el agujero se troqueló algo más bajo en la carcasa, para poder distinguirlos de los disquetes de alta densidad.

Unidades de disquete y sus controladoras

Una unidad de disquete se compone de una mecánica, que se encarga de que el disquete gire a 300 revoluciones por minuto (360 en High Density, 5 1/4") y del movimiento del cabezal. Además, la unidad de disquete contiene algo más de electrónica en forma del llamado separador de datos, que convierte la alternancia de tensión al leer el disquete en una corriente de datos binaria.

El auténtico control de la unidad se realiza, sin embargo mediante la controladora de disquetes que va separada, y que habitualmente es una tarjeta de expansión que se encuentra en una de las ranuras de expansión del ordenador. Mientras que la controladora de disquetes en los PC y XT sólo controlaba el acceso a las diferentes unidades de disquete, y podía gestionar cuatro de ellas, la controladora de disquetes del AT se encuentra junto con la controladora de discos duro en la misma placa. Por ello también se habla de una controladora combinada, con la que se pueden controlar, además de las dos unidades de disquete, dos discos duros.

En el centro de una de estas controladoras de disquetes existe un circuito de control del tipo NEC PD765 (o similar, de otro fabricante, que sea compatible, ya que sólo la compatibilidad de registros con el chip de NEC, que IBM utilizó desde el principio en sus unidades de disquete, garantiza la cooperación sin problemas con la ROM BIOS. Al fin y al cabo, es la ROM BIOS el que controla el acceso a la unidad de disquete mediante este Chip.

A pesar de que es posible adaptar al ROM BIOS para que trabaje con otros controladores de disquetes, pero la mayoría de fabricantes de sistemas ROM BIOS se atienen al NEC PD765, que se ha establecido como estándar.

Pero exactamente esto es lo que trae problemas con las nuevas unidades de disquete ED, cuyo doble tamaño de pista (36 en vez de 18 sectores), y con ello la duplicación de la tasa de transferencia no le son conocidos al BIOS. La única solución la representa una ampliación de la ROM de la controladora correspondiente, ya que durante el arranque del sistema se «engancha» en el ROM BIOS y realiza por ella mismo el acceso a estas unidades, sin utilizar el ROM-BIOS. Pero esto sólo funciona hasta que el ordenador se conmuta al Protected Mode, en el que el BIOS de todas formas carece de valor, ya que únicamente está diseñado para el Real-Mode.

Los sistemas operativos como UNIX o OS/2 deberán encargarse de momento de la programación de la controladora de disquetes, y fallar en sucesivas unidades, a no ser que existan controladores (drivers) especiales. Por ello es de esperar y desear que se cree cuanto antes un estándar BIOS para el acceso a las unidades ED. Y esto es tanto mas cierto cuanto Windows en el futuro asumirá cada vez más funciones del BIOS, de modo que esta problemática gana en importancia con la expansión de las unidades ED de 2,88 MB, para el usuario.

Acceso a disquetes con el BIOS

Para acceder a disquetes, el BIOS contiene toda una serie de funciones, que se pueden llamar a través de la interrupción 13h. Esta interrupción funciona como interfaz al servicio del disco duro del BIOS. Siempre que sea posible, funciones semejantes de disquetes y discos duros deben ocupar un número de función idéntico. La diferenciación se realiza en este caso mediante la indicación de unidad, que se ha de cargar antes de la llamada de la función al registro DL.

Para unidades de disquete se espera el valor 0 (unidad A) o 1 (unidad B). Las pocas controladoras de disquete que soportan cuatro unidades de disquete, disponen habitualmente de una ampliación del BIOS, que también acepta los valores 2 y 3 para las dos unidades de disquete adicionales. Los discos duros, por el contrario, se representan con los valores 80h y 81h.

Al observar las diferentes funciones, se ha de distinguir entre el BIOS del PC/XT y el BIOS del AT, ya que con las unidades de disquete MF se tuvieron que introducir diferentes funciones del BIOS en el AT, que acepten la funcionalidad ampliada de estas unidades.

Esto se puede ver en la siguiente tabla, que lista las funciones de disquete de la interrupción BIOS 13h.

Nro.	Tarea	PC/XT	AT
00h	Reset	Si	Si

01h	Leer estado	Si	Si
02h	Leer	Si	Si
03h	Escribir	Si	Si
04h	Verificar	Si	Si
05h	Formatear	Si	Si
08h	Averiguar formato	Si	Si
15h	Determinar tipo de unidad	No	Si
16h	Determinar cambio de disco	No	Si
17h	Fijar formato de disco	No	Si
18h	Fijar formato de disco	No	Si

Por cierto, no hay nada en contra de implementar las funciones AT también en PC y XT, siempre y cuando dispongan de unidades MF. Si esto ha ocurrido en la práctica con uno de los innumerables fabricantes de PC, no sabríamos decirlo.

Si echa un vistazo a la tabla anterior, seguramente se habrá fijado en las funciones 17h y 18h, que realizan la misma tarea. Esto no es un error de imprenta, sino que realmente es así. Ello demuestra las dificultades que ha dejado tras de sí la introducción de las unidades de disquete de 31/2" High Density en el mundo de las ROM-BIOS. Ya que la función 17h simplemente no estaba preparada, y por ello tuvo que sustituirse por una función nueva. Pero de ello hablaremos de ello más adelante, en relación con el formateo de los disquetes.

El estado de la unidad

Otra cosa en común entre las diferentes funciones es, además del paso del número de unidad en el registro DL, la devolución de un código de estado o error en el registro AH. Un error se representa por un valor distinto de 0 y se acompaña de la bandera de carry activada.

Tabla : Códigos de error y estado de las funciones de disquete BIOS

Código	Significado
00h	No hay error
01h	Núm. de función no permitido
02h	No se encontró marca de dirección
03h	Intento de escritura en disquete protegido
04h	Sector especificado no se encuentra
06h	Disco se cambió
08h	Rebase DMA
09h	Transmisión de datos más allá del límite de segmento
10h	Error de lectura
20h	Error de la controladora de disquetes
40h	Pista no encontrada
80h	Error Time-Out, unidad no responde

El estado de disquetes también se puede obtener en cualquier momento con ayuda de la función 01h, que no tiene otra tarea, que devolver precisamente esta información. Coloque antes de la llamada de la función el número de función 01h en el registro AH, y el número identificativo de la unidad deseada en el registro DL. Después de la llamada de la función, obtendrá el estado, como siempre, en el registro AH.

Reset de la unidad de disquete

Si después de la llamada de una función de disquete reconoce, con ayuda del código de error o estado, que ha aparecido un error, debería realizar un reset con ayuda de la función 00h. Aparte del número de función (AH) y de la denominación de la unidad (DL) no espera otros parámetros, pero a su vez devuelve de nuevo el estado actual de la unidad en AH.

Por cierto que es igual si indica como denominación de unidad 0 o 1, ya que por norma se realiza un reset en todas las unidades de disquete. En cambio no es indiferente el valor en el registro DL, ya que al indicar un valor superior a 80h, se realiza un reset en los discos duros conectados, en vez de en las unidades de disquete.

Averiguar el tipo de unidad

No siempre un programa sabe qué tipo de unidad tiene ante sí, y qué tipos de formato de disquete se soportan. Con ayuda de las funciones 08h y 15h se puede obtener esta información. La primera ya se puede encontrar en el BIOS del PC/XT, y sirve para diferenciar entre los diferentes formatos de disquete y unidad. En su llamada espera simplemente el número de unidad en el registro DL. Devuelve toda una serie de informaciones, como muestra la siguiente figura.

Como es habitual, después de cada llamada de función la bandera de carry está activada, y en el registro AH se puede encontrar un código de error, si la llamada de función ha fallado. Con ello se puede averiguar si hay una determinada unidad instalada, y, por lo tanto se puede demostrar la posible presencia de una segunda, tercera o incluso cuarta unidad.

Tabla : Informaciones de retorno de la función 08h

Registro	Informaciones
BL	Tipo de unidad
	01h = 51/4", 360 KB
	02h = 51/4", 1,2 MB

03h = 31/2", 720 KB

04h = 31/2", 1,44 MB

DH Número mayor de cara (siempre 1)
CH Número mayor de pista
CL Número mayor de sector
ES:DI Puntero a DDPT

Especialmente interesante es el valor en el registro BL, que no sólo indica el formato de la unidad (31/2" o 51/4"), sino también el formato de los disquetes (Double Density o High Density). Pero con ello no describe el formato del disquete insertado, sino simplemente el mayor posible.

Esta información se obtiene de la RAM CMOS, en la que se guardaron estas informaciones durante la configuración del ordenador. La distribución de números de código está estandarizada, pero aún no recoge las nuevas unidades ED de 3 1/2". Pero suponemos, que estas unidades en un futuro figurarán bajo el número de código 05h, ya que otra cosa no sería lógica.

Del código de unidad resulta, por consiguiente, la cantidad de sectores, pistas y cabezales, pero estas informaciones se devuelven de nuevo explícitamente en los registros CH/CL, DH/DL.

En la DDPT, que se referencia mediante un puntero en la pareja de registros ES:DI, se trata de la llamada Disk-Drive-Parameter-Table, que en lo subsiguiente nos la encontraremos más veces. Contiene parámetros que el BIOS necesita para la programación de la controladora de disquetes. Puede encontrar una descripción de esta tabla más adelante.

Una funcionalidad algo distinta la tiene la función 15h, que sólo se soporta por los AT y sus unidades MF ya que, al contrario que las unidades PC/XT son capaces de reconocer un cambio de disco. Para muchos programas esto es especialmente importante, ya que se ha de evitar que el usuario cambie un disco inadvertidamente. Esto es especialmente importante para el DOS, que durante el primer acceso a un disquete lee la llamada tabla FAT, para averiguar con su ayuda qué partes del disquete ya están ocupados con archivos y cuales quedan libres. Si entonces se cambia un disquete inadvertidamente, el DOS sigue trabajando con la «Tabla de alojamiento» (FAT) antigua, y por ello podría acceder a sectores, que en el nuevo disquete podrían estar ocupados por archivos. Naturalmente también puede ocurrir que se vuelva a colocar el mismo disquete, y que por lo tanto no sea necesario leer de nuevo la FAT. Puede ver que la problemática es algo más compleja de los que se podría pensar a primera vista, pero este problema afecta principalmente al nivel DOS, y en este punto no es demasiado importante.

El BIOS sólo dispone de los métodos de ayuda, con los que el DOS puede determinar si en la «caza» del cambio de disquete es soportado por la unidad. La función 15h, análogamente a la función 08h, sólo espera que se le pase el número de función en el registro AH y la denominación de unidad en el registro DL.

Como valor de retorno devuelve un código según la siguiente tabla, que aparte del estado deseado muestra, si la unidad especificada está presente y si por casualidad se trata de una unidad de disco duro.

Tabla: Informaciones de retorno de la función 08h

Registro Informaciones

BL Tipo de unidad
01h = 51/4", 360 KB
02h = 51/4", 1,2 MB
03h = 31/2", 720 KB
04h = 31/2", 1,44 MB
DH Número mayor de cara (siempre 1)
CH Número mayor de pista
CL Número mayor de sector
ES:DI Puntero a DDPT

Leer sectores de disquete

Uno de los servicios elementales que ha de ofrecer el BIOS, es la lectura de los sectores individuales de un disquete. La función 02h sirve a este propósito. Se le ha de comunicar para ello, cual o cuales sectores han de ser leídos, ya que en una pasada se pueden leer directamente varios sectores, siempre que sean parte de una pista, y sean contiguos. Los siguientes registros se han de cargar antes de la llamada de la función con la dirección del (primer) sector a leer.

Tabla : Registro durante la llamada de la función 02h

DL Número de la unidad
DH Cabezal de lectura/escritura (0 = arriba, 1 = abajo)
CL Número de sector (1 a N)
CH Número de pista (0 a N-1)
AL Número de sectores a leer

Ya que los datos no se transfieren automáticamente en una zona de memoria fija, se debe especificar la dirección del buffer en la pareja de registros ES:BX. Como siempre, ES acoge la dirección de segmento del buffer y BX la dirección de offset. Si estos parámetros han sido cargados correctamente, para la llamada de función sólo se necesita la denominación de unidad en el registro DL y el número de función en el registro AH.

Después de un acceso de lectura, la función devuelve en el registro AH el estado de error y en el registro AL el número de sectores leídos. La bandera de carry activada señala la aparición de un error.

La lectura de sectores individuales también se puede utilizar incorrectamente para obtener el formato de un disquete, siempre y cuando se esté tratando con una unidad MF, por lo que se ha de contar con disquetes Double Density y HD. Un acceso de lectura a un número de sector mayor que 9 nos aclara las cosas, ya que en los disquetes Double Density éste no existe, y por lo tanto la función nos devolverá un error. El número de pista no tiene importancia, aunque debería ser inferior a 40, ya que en caso contrario se presupone el formato.

Aunque se desee un error en este caso, no se debería interrumpir una operación en esta situación inmediatamente. Ya que todas las llamadas de lectura, escritura y formateo se deberían repetir un mínimo de tres veces, hasta que uno se da por vencido y se parte de un error. Frecuentemente una operación falla la primera vez, pero funciona si se intenta una segunda o tercera vez. Esto tiene que ver con que el cabezal de lectura/escritura no estaba correctamente posicionado durante el primer intento, o que la unidad aún no se había sincronizado correctamente al cambio de flujo.

Sin embargo no se ha de temer un error en los datos devueltos, ya que la unidad guarda con cada sector un valor de comprobación, con ayuda del cual puede comprobar la corrección de las informaciones leídas.

Escribir sectores de disquetes

Para escribir en sectores individuales se utiliza la función 03h, cuya ocupación de registros es muy similar a la de la función 02h. La única diferencia con la función 02h consiste en que el buffer indicado ya debe haber sido cargado, con los datos a escribir, antes de la llamada de la función y, concretamente, del programa invocador.

Tabla : Registros en la llamada de la función 03h

AL	Número de sectores a escribir
DL	Número de la unidad
DH	Cabezal de lectura/escritura (0 = abajo, 1 = arriba)
CL	Número de sector (1 a N)
CH	Número de pista (0 a N-1)
AL	Número de sectores a leer
ES:BX	Puntero al buffer con los datos

Verificar sectores de disquete

Si quiere comprobar si los datos se han transmitido correctamente al disquete, puede utilizar la función 04h. Sin embargo como se suele creer no se comparan los datos en la memoria con los datos del disquete, sino que simplemente se comprueba la correcta transferencia de datos con ayuda del valor CRC. CRC significa «Cyclical Redundancy Check», y representa un método de control muy fiable, en el que los valores de cada uno de los Bytes en un sector se operan con complejas funciones matemáticas en una suma. Como la mayoría de unidades de disquete son muy fiables, y trabajan bien, esta rutina es considerada innecesaria por la mayoría de programadores, y no se emplea. Incluso el DOS sólo emplea esta función durante la escritura de datos, si primero se introdujo el comando VERIFY ON.

La ocupación de registros se parece a la ocupación de registros de las funciones 02h y 03h en cuanto a llamada y retorno, con la diferencia que no es preciso indicar una dirección de buffer.

Formatear sectores de disquetes

Naturalmente también se pueden formatear los disquetes con ayuda del BIOS, pero esto tiene un problema que surgió con la introducción de las unidades MF al mundo de los PC, y con ella el soporte del formato High Density: la selección del formato de disquetes, ya que el BIOS ha de saber, para poder formatear un disquete, cuantos sectores por pista y cuantas pistas se han de «instalar».

Así que se trata del formato del disquete una información, que normalmente se puede averiguar realizando un acceso al décimo, decimoprimer o decimosegundo sector de una pista, como hemos mostrado anteriormente. Pero al formatear, este truco no sirve, ya que los sectores que se han de leer, aún no existen en este punto.

Esta es también la razón, por la que el usuario ha de especificar el parámetro /4 para formatear un disquete Double Density en una unidad 5 1/4" bajo DOS. Ya que el DOS necesita esta información, para ajustar el formato deseado. El programa FORMAT utiliza una función del BIOS, que se ha de llamar siempre antes del primer formateo de una pista: la función 18h.

Esta función se introdujo en el AT, y sustituye la antigua función 17h, que hasta entonces se había usado para estos propósitos. Como parámetros, la función 18h espera, aparte del número de función en el registro AH, el número deseado de pistas en el registro CH y el número de sectores por pista en el registro CL. Además, naturalmente también espera el número de unidad en el registro DL.

Si el formato indicado se soporta por la unidad correspondiente, la función vuelve con la bandera de carry borrada, y le devuelve al invocador un puntero al DDPT en la pareja de registros ES:DI, que se necesita para el formateo bajo el formato especificado.

Lo que ocurre con el DDPT se describe en el siguiente apartado. En este momento simplemente vamos a determinar que el puntero devuelto se ha de llevar al vector de interrupciones 1Eh, en la que el BIOS guarda el puntero a la DDPT actual.

Si el formato deseado se ha ajustado y la DDPT devuelta se ha instalado, por fin puede comenzar el proceso de formateo en sí.

Para ello se utiliza la función 05h, con cuya ayuda se formatea una pista completa cada vez. Los diferentes sectores se pueden formatear con 128, 256, 512, o incluso 1024 Byte por sector, pero sólo se puede elegir el formato de 512 Byte, si el disquete se ha de utilizar posteriormente bajo DOS, ya que DOS sólo soporta este tamaño de sector.

Al llamar la función 05h, el registro AL debe contener el número de sectores de la pista a formatear. En el registro CH se espera el número de la pista a formatear, que se debe encontrar entre 0 y 39 o entre 0 y 79. El número de la unidad de disquetes se debe grabar en el registro DL, y la cara de disquete en el registro DH.

Tabla : Registros al llamar la función 05h

AL Número de sectores por pista
DL Número de la unidad
CH Número de la pista
DH Cara

ES:BX Puntero a la tabla de formatos

Aparte de estas indicaciones, que también se esperan en las funciones para leer, escribir y verificar un sector, para la función de formateo se necesita un campo con las características de formateo. La dirección de este campo se espera como puntero FAR en la pareja de registros ES:BX.

Esta tabla representa una especie de Array, que para cada sector a formatear debe de contener una entrada, que se compone de cuatro Byte.

Tabla: Entradas en la estructura del array de formato

Offset	Significado
0	Pista a formatear
1	Cara del disquete (siempre 0 en disquetes de 1 cara): 0 = Anverso 1 = Reverso
2	Número del sector
3	Número de Byte en este sector 0 = 128 Byte 1 = 256 Byte 2 = 512 Byte 3 = 1024 Byte

A pesar de que el número de la pista a formatear y la cara de disquete ya se le han de especificar a la función 05h durante su llamada, aquí se han de listar una segunda vez. Los sectores se crean físicamente en el orden de las entradas de la tabla, donde es posible de darle al primero el número 1, y al segundo el número 7. Ya que el número lógico del sector se guarda en el disquete en la cabecera de cada sector, para que la unidad pueda localizar un sector buscado.

Ya que el BIOS no determina el número lógico de sector por sí misma, se pueden desplazar con respecto a los sectores físicos, con lo que se genera lo que se llama «interleaving». Pero habitualmente este método es sólo empleado por los discos duros. Pero en el formateo de los disquetes, esto no aporta ninguna ventaja, y por ello les recomendamos una numeración seguida de los diferentes sectores al crear el array de formato.

También la cantidad de Byte que puede acoger cada sector no ha de ser unitaria, ya que se define individualmente por cada sector. Precisamente con esto, y también en conjunto con los otros parámetros de esta tabla, se puede desarrollar una protección de copia, como se muestra a continuación.

Cómo se pueden formatear los disquetes con ayuda de las funciones 18h y 05h en la práctica, se demuestra con un programa de ejemplo al final de este capítulo.

La Disk-Drive-Parameter-Table

Aparte de las indicaciones sobre el formato físico de un disquete, el BIOS necesita, para la programación de la controladora de disquetes, toda una serie de informaciones. Se guardan en la llamada Disk-Drive-Parameter-Table (abreviada: DDPT), que a lo largo de este capítulo ya ha sido mencionada.

Para cada unidad y para cada formato de disquete ya existe una de estas tablas en la ROM BIOS. Pero es posible el instalar una DDPT propia, ya que el BIOS siempre referencia la DDPT actual mediante un puntero FAR, que se guarda en las posiciones de memoria que normalmente ocupa la interrupción 1Eh. El uso de estas posiciones de memoria se hace posible, ya que la interrupción 1Eh no se emplea por el hardware del PC, ni por el DOS, ni tampoco sirve como interfaz a determinadas funciones del BIOS.

El DOS hace uso de la posibilidad de instalar una DDPT propia, modificando determinados valores de esta tabla con respecto al BIOS. Detrás se esconde la idea de poder acelerar los accesos a los disquetes de esta forma, lo que ciertamente es posible, ya que algunos valores de la tabla se dimensionaron de forma muy generosa. Pero esto naturalmente genera la pregunta de qué es lo que hay exactamente en esta tabla.

La tabla en sí se compone de 11 Byte, como muestra la siguiente ilustración. No todos los parámetros se pueden modificar, pero en los marcados en gris hay algo de margen de juego.

Tabla : Estructura de la Disk-Drive-Parameter-Table

Offset	Significado	Tipo
00h	Step-Rate y Head-Unload-Time	1 BYTE
01h	Head-Load-Time	1 BYTE
02h	Tiempo de paro del motor de la unidad	1 BYTE
03h	Tamaño de sector	1 BYTE
04h	Sectores por pista	1 BYTE
05h	Longitud de GAP3 al leer/escribir	1 BYTE
06h	DTL (Data Length)	1 BYTE
07h	Longitud de GAP3 al formatear	1 BYTE
08h	Byte de relleno para formatear	1 BYTE
09h	Head-Settle-Time	1 BYTE
0Ah	Tiempo de arranque del motor de la unidad	1 BYTE

El primer byte DDPT está dividido en dos y acoge en sus dos Nibbles la Steprate (bit 47) y el Head-Unload-Time (bit 0-3). La Steprate describe el tiempo que se le deja a la controladora para adelantar el cabezal de lectura/escritura una pista. Se cuenta en milisegundos, donde el valor 0Fh está por 1 ms, 0Eh por 2 ms, 0Dh por 3 ms, etc. El Head-Unload-Time describe el tiempo que se le deja al cabezal para levantarse del disquete, por ejemplo cuando se va a cambiar de pista. Se indica como un factor de 16 ms. El valor por defecto 0Fh (240 ms) está elegido de forma bastante generosa, y habitualmente se puede bajar.

En el segundo byte se encuentra el Head-Load-Time, el tiempo, que se le deja al cabezal de lectura/escritura para bajar a una pista. Ocupa los bits 1 a 7, y se indica como factor de 2 ms. Ya que durante el acceso a disquete se espera bastante más a la aceleración del motor de la unidad hasta su velocidad normal, para el Head-Load-Time se especifica normalmente un valor muy bajo (1 o 2). El bit 0 en este byte se ocupa normalmente por el llamado DMA-Flag, que decide sobre la utilización del canal DMA mediante la unidad de disquete. Este bit siempre ha de ser 0.

El tercer byte indica el tiempo de espera del motor de la unidad de disquete, después de una operación de disquete. Este es el tiempo que pasa hasta que se desconecta el motor de la unidad, si no se realizan más operaciones de disquete. Ya que la aceleración del motor cuesta mucho tiempo, y por ello tiene sentido no apagarlo inmediatamente después de cada acceso a disquete. El valor en esta posición de memoria se refiere al reloj del sistema de aprox. 18 ticks por segundo (1 tick corresponde a unos 55 ms), de modo que un valor de 18 corresponde a un tiempo de espera de aproximadamente un segundo. El valor por defecto es de 25h, lo que corresponde a 2 buenos segundos.

El valor en el siguiente byte indica los Bytes por sector, con los que se trabaja en una operación de lectura o escritura. Corresponde a los valores para el formateo de un sector, de modo que generalmente contiene el valor 3 para 512 Bytes por sector. Si se quieren leer o escribir sectores con otros tamaños de sector, primero se ha de escribir el valor correspondiente en esta posición de memoria.

En la dirección de offset 04h se puede encontrar el número de sectores máximo por pista, que depende del formato de disquetes elegido.

Los siguientes tres Bytes se refieren a la codificación y decodificación de las informaciones de sector, que se guardan adicionalmente a los datos en el disquete. ¡En estos valores no se debería tocar nada!

De nuevo interesante es el bit en la dirección de offset 08h. Acoge el código ASCII del carácter, con el que se rellena un sector durante su formateo. Ya que los sectores, durante su formateo, no solo se crean, sino que directamente se rellenan con un contenido constante. Aquí se emplea el signo de división (Código ASCII 246).

Después de que el cabezal de lectura/escritura haya ido de una pista a otra, necesita un poco de tiempo de descanso, para que las vibraciones derivadas de este movimiento de detengan. Sólo entonces está de nuevo en disposición de realizar los siguientes accesos a disquete de forma adecuada. Cuanto tiempo se le deja, está en el byte con la dirección de offset 09h, como factor de 1 ms. El valor por defecto es de 25 ms.

La última entrada de la DDPT indica la duración, en la que el motor del disquete alcanza su velocidad de trabajo. El valor en esta posición de memoria es un factor de 1/8 segundo. También aquí el DOS le pide más al motor de la unidad de disquete, que el BIOS. Simplemente prevé 1/4 de segundo de tiempo de espera respecto al 1/2 segundo del BIOS.

Acceso a discos duros con ayuda del BIOS

Este capítulo le describe las diferentes funciones que contiene el BIOS para el acceso a discos duros. Antes de entrar en detalles, queremos prevenirles de no jugar demasiado con estas funciones. Al contrario que en una unidad de disquete, en la que puede insertar un disco sin utilizar, en los discos duros, como es sabido, esto no es posible. Y así, utilizando inadecuadamente las funciones de escritura o de formateo, puede provocar una pérdida de datos irrecuperable. A causa de la estructura que DOS deposita en el disco duro, un sólo sector destruido puede bastar para hacer desaparecer de pronto todos los directorios y archivos, ya que DOS ya no sabe donde están en el disco.

Si quiere «probar» las funciones del BIOS, debería hacer en cualquier caso un Backup completo de todo el disco duro o bien utilizar un ordenador sin usar, siempre y cuando tenga alguno disponible. Sólo con ello puede prevenir la pérdida de datos, ya que incluso la utilidad de disco más probada puede no recuperar lo que puede llegar a perder.

La interrupción de disco duro del BIOS

Como ya se mencionó en la descripción del BIOS de disquetes, el disco duro y los disquetes comparten la interrupción 13h. A pesar de que las diferentes funciones de las que se dispone para disquetes y discos duros, son en cuanto a su tarea, idénticas, el disco duro tiene un control totalmente distinto del de las unidades de disquetes. Por esta razón el BIOS contiene un módulo separado para controlar el disco duro, y otro para el control de las unidades de disquete.

Si se llama la interrupción 13h, el número de dispositivo en el registro DL decide sobre si se accederá a una unidad de disquete, o una de disco duro. El valor 80h está por el primer disco duro, 81h por el segundo, y habitualmente no se pueden direccionar más de dos unidades de disco duro mediante el BIOS.

Las funciones del BIOS del disco duro sólo existen desde la introducción del XT, ya que en el PC original no estaban contenidas. ¿Y para qué?, ya que en el año 1981 no se pensaba aún en discos duros para microordenadores. Con el tiempo, mejor dicho: con la introducción del AT y los modelos PS/2 de IBM, se han añadido algunas funciones, como muestra la siguiente tabla. Véase la tabla.

Tabla : Interrupción de funciones de disco duro

Función	Tarea	desde...
00h	Reset	XT
01h	Leer estado	XT
02h	Leer	XT
03h	Escribir	XT
04h	Verificar	XT
05h	Formatear	XT
08h	Obtener formato	XT
09h	Adaptar unidades ajenas	XT

0Ah	Lectura ampliada	XT
0Bh	Escritura ampliada	XT
0Ch	Mover cabezal lectura/escritura	XT
0Dh	Reset	XT
0Eh	Test de lectura de la controladora	sólo PS/2
0Fh	Test de escritura de la controladora	sólo PS/2
10h	¿Unidad preparada?	XT
11h	Recalibrar la unidad	XT
12h	Controller-RAM-Test	solo PS/2
13h	Test de unidad	solo PS/2
14h	Diagnóstico de controladora	XT
15h	Determinar el tipo de unidad	AT

Ya que el acceso al disco duro con el BIOS no tiene importancia en el trabajo de programación diario, en este capítulo vamos a limitarnos a la descripción de las funciones más importantes. Puede obtener otras informaciones en el apéndice de este libro, en el que se describen todas las funciones con sus parámetros de entrada y salida.

El código de estado

Todas las funciones de disco duro tienen en común la circunstancia, que después de su llamada comunican a través de la bandera de carry si pudieron realizar su tarea, y si ocurrió un error. Si esto es así, la bandera de carry está activada y en el registro AH se encuentra un código de error.

Los diferentes códigos tienen el siguiente significado.

Tabla : Códigos de error al llamar la interrupción de disco del BIOS 13h para acceder a la unidad de disco duro

Código	Significado
00h	No hay error
01h	Número de función o unidad ilegal
02h	No se encontró marca de direcciones
04h	Sector direccionado no se encontró
05h	Error al reset de la controladora
07h	Error al inicializar la controladora
09h	Transf. de datos más allá del límite de segmento
0Ah	Sector defectuoso
10h	Error de lectura
11h	Error de lectura corregido con EEC
20h	Error de la controladora de disquetes
40h	Pista no encontrada
80h	Error de Time-Out, unidad no responde
AAh	Unidad no preparada
CCh	Error de escritura

Por norma, al aparecer un error (excepto el error 1) debería realizar un reset para comenzar, e intentar otra llamada de la función. En muchos casos, ya no se indicará error.

Si durante la llamada de una función de lectura apareciera el error 11h, los datos leídos no necesariamente han de ser defectuosos. Sino que este error indica, que se descubrió un error de lectura, pero que se pudo corregir con ayuda del algoritmo ECC («Error Correction Code»). En este procedimiento se trata de un método, que es parecido al sistema CRC de las unidades de disquetes. Los diferentes Byte de un sector se pueden operar mediante una complicada fórmula matemática. El resultado de esta operación se añade en forma de 4 Bytes al sector en el disco. Si se determina un error de lectura, este error, con ciertos límites se puede corregir con ayuda del resultado ECC guardado.

Llamada de las funciones de disco duro

Otra cosa en común entre las diferentes funciones consiste en la utilización de los registros del procesador para la transmisión de datos. Por norma, el número de función se pasa en el registro AH. Si se pide el número de la unidad de disco duro a direccionar, esta siempre se ha de pasar en el registro DL. El valor 80h siempre indica la primera unidad de disco duro, 81h la segunda unidad de disco duro. El número del cabezal de lectura/escritura, y con ello indirectamente el numero del disco direccionado (cada disco dispone de dos cabezales de lectura/escritura) se anota en el registro DH.

El número de cilindros lo acoge el registro DH. Pero como con este registro de 8 bits sólo se pueden direccionar 256 cilindros, y ya el disco duro de un XT dispone de 306 cilindros, este registro no es suficiente para la indicación del número de cilindro. Por esta razón los bits 6 y 7 del registro CL se utilizan adicionalmente para especificar el número del cilindro. Forman los bits 8 y 9 del número de cilindro, de modo que se pueden direccionar un total de 1024 cilindros (con los números 0 hasta 1024). Los bits 0 a 5 del registro CL no se quedan sin utilizar, ya que indican el número del sector a direccionar (están numerados por cilindro de 1 a 17). Si se quiere acceder simultáneamente a varios sectores, el registro AL acoge el número de estos sectores. En operaciones de lectura y escritura naturalmente también se ha de indicar la dirección de un buffer, del que se han de tomar los datos, o al que se han de transferir. es este caso, el registro ES acoge la dirección de segmento y el registro BX la dirección de offset del buffer.

Reset de la controladora de disco duro

Una función que no necesita estas indicaciones, es la función 00h, que, igual que la función 0Dh realiza un reset en la controladora. Esta función es imprescindible después de la aparición de un error, ya que antes del siguiente acceso a los datos se ha de realizar un

reset después de aparecer un error. La indicación de la unidad de disco duro en la que se debe realizar el reset, se especifica como siempre en el registro DL.

Obtener el estado de la unidad de disco duro

Con ayuda de la función 01h se puede obtener el estado de la unidad de disco duro, que se indica después de cada operación. También aquí se ha de guardar el número de la unidad cuyo estado se quiere obtener, en el registro DL.

Leer sectores del disco duro

Para leer uno o varios sectores, sirve la función 02h. Mediante una llamada única de esta función, se pueden leer un máximo de 128 sectores. Esta limitación proviene de que los datos de la controladora del disco duro se transfieren directamente a la memoria RAM con ayuda de un circuito DMA, es decir sin pasar por el procesador. Pero el circuito DMA sólo puede transferir 64 KB en una sola vez. Y esto sólo puede hacerlo en un segmento de datos. Por esta razón es importante, que el buffer completo, cuya dirección se pasa en ES:BX, quepa en los 64 KB a partir de la dirección de segmento especificada en ES, ya que de lo contrario, el circuito DMA dará error.

Si quieren leer varios sectores con esta función, también es interesante saber qué sectores se han de leer por la función. Para comenzar naturalmente los sectores con números ascendentes dentro del cilindro indicado en el cabezal especificado. Pero si se ha leído el último sector de un cilindro y se deben leer varios sectores, no se continua la lectura con el primer sector del siguiente cilindro en el cabezal indicado. Sino que el siguiente sector a leer es el primer sector del mismo cilindro, pero en el siguiente cabezal. Sólo cuando de esta forma se ha llegado al último cabezal, y es necesario leer más sectores, el proceso de lectura se continua en el primer sector del siguiente cilindro en el primer cabezal.

Todo esto también es cierto para la función de escritura, que lleva el número de función 03h.

Verificar sectores de disco duro

Con ayuda de la función 04h se pueden verificar los diferentes sectores de un cilindro. Pero los datos del disco no se comparan con los datos en la memoria (por ello no es necesario especificar una dirección de buffer en ES:BX), sino que se comprueba con ayuda del número ECC, si los Byte guardados durante su operación con el algoritmos ECC vuelven a dar el resultado almacenado. La cantidad de sectores a verificar se indica de nuevo en el registro AL.

Formatear cilindros del disco duro

Antes de acceder a un disco duro, naturalmente ha de ser formateado. Para esta tarea se dispone de la función 05h. De forma parecida que en la función para formatear un disquete, a la función se le han de pasar, aparte de los número de cabezal y cilindro, la dirección del buffer en la pareja de registros ES:BX. Este buffer ha de tener al menos 512 Byte, aunque sólo se empleen los primeros 34 Byte.

Para cada uno de los 17 sectores a formatear, contiene dos Byte. El primer byte indica cada vez si el sector está bien. Ya que antes de la llamada de la función se parte de que cada sector esté bien, en este byte siempre se graba el valor 0. El segundo byte acoge el número (lógico), que se le ha de dar al sector correspondiente. El BIOS coge de los dos primeros Byte de la tabla informaciones sobre el primer sector físico del cilindro, de los Byte 3 y 4 las informaciones sobre el segundo cilindro físico, etc. Así que mientras el orden físico del los cilindros ya esta determinado, el orden lógico de los sectores se puede determinar con ayuda de los dos Byte de esta tabla.

Ahora se podría pensar que sería lo más útil el asociar a cada sector lógico el número del sector físico. Pero esto no es así. Por razones de un acceso lo más rápido posible, normalmente se desplazan los sectores lógicos con respecto a los físicos; en inglés esto se denomina como método Sector-Interleaving.

Durante la llamada de función, el BIOS escribe en el primer byte de la identificación de sectores en la tabla un valor, que le indica al programa invocador si el sector está bien, o si por lo contrario no debería escribirse en el a causa de una magnetización deficiente. El valor 0 significa OK y el valor 128 significa estropeado. Aparte de los registros antes mencionados, en esta función el registro AL acoge el número de sectores a procesar.

Averiguar valores límite del disco duro

Ya que no existe un formato uniforme para los discos duros, al contrario que en los disquetes, en determinadas situaciones es importante para un programa el averiguar los «valores límite» de la unidad de disco duro conectada. Puede utilizar la función 08h, a la que se le ha de pasar en la llamada, aparte del número de función como siempre el número de unidad de disco duro en el registro DL. Después de la llamada de función, el registro DL contiene el número de discos duros conectados. Aquí son factibles los valores 0, 1 y 2. Aparte de ello, el registro DH contiene el número de cabezales de lectura/escritura. Pero como en los cabezales siempre se cuenta desde 0, el valor 7, por ejemplo, significa que hay 8 cabezales. El número de cilindros se devuelve, al igual que en las otras llamadas de función, en el registro CL (bits 0-7 del número de cilindros) y en los dos bits superiores del registro CL (bits 8-9 del número de cilindros). También aquí se cuenta desde 0. La última información encuentra espacio en los 6 bits bajos del registro CH. Indica el número de sectores por cilindro, donde excepcionalmente se cuenta desde 1.

Adaptar discos duros desconocidos

Todo BIOS contiene ya los valores límite de un número mayor o menor de discos duros, para que estos se pueden ajustar durante la configuración del ordenador, y consiguiendo con ello que el BIOS este en condiciones de configurar estos discos.

Si se quieren hacer funcionar discos duros desconocidos con ayuda del BIOS, naturalmente se le han de dar a conocer las características especiales de este disco duro de otra forma. Por esta razón utiliza la interrupción 41h para la unidad de disco duro 0 y la interrupción 46h para la unidad de disco duro 1 como puntero a una tabla. Esta tabla tiene un formato definido por el BIOS, y describe la unidad de disco duro conectada. Normalmente los dos vectores de interrupción acogen punteros a la tabla guardada en el BIOS, pero se pueden cambiar por software a otra tabla, desconocida de momento por el BIOS.

Sólo que, después de instalar semejante tabla y después de la inicialización del vector de interrupción correspondiente no está todo. Además de ello, ha de hacer que el BIOS se ajuste a esta tabla. Para ello sirve la función 09h, que aparte del número de función espera

el número de la unidad de disco duro a adaptar (80h o 81h) en el registro DL. Pero esto normalmente se realiza con un pequeño controlador, que muchos fabricantes de discos duros ya ponen a disposición.

Leer sector de disco duro ampliado

Otras dos funciones de lectura y escritura se obtienen con las funciones 0Ah y 0Bh. Se diferencian de las funciones 02h y 03h solamente en que aparte de los 512 Byte de datos por sector adicionalmente leen los cuatro Byte ECC que hay al final de cada sector, y los guardan. Ya que con ello cada sector tiene prácticamente 516 Byte en vez de 512, no se pueden leer o escribir 128 sectores sino solamente 127.

Si la unidad de disco duro cuyo número está contenido en el registro DL está lista en este momento para ejecutar comandos, se puede comprobar con ayuda de la función 10h. La respuesta a esta pregunta la devuelve la función mediante la bandera de carry. Si está activada, la unidad no está lista, y uno de los códigos de error habituales se encuentra en el registro AH.

Recalibrar disco duro

Para recalibrar una de las dos unidades de disco duro se puede llamar la función 0Bh. También ella devuelve después de la llamada de función, el número de unidad en el registro DL según las reglas habituales de estado de error.

Auto-test de la controladora de disco duro

Para hacer que la controladora realice un auto-test, se puede llamar la función 14h. Si la controladora está OK, vuelve con la bandera de carry borrada.

La última función de disco duro, que se puede alcanzar mediante la interrupción 13h, es la función 15h, pero sólo está disponible en los AT, y no en los XT. Sirve para averiguar el tipo de una unidad determinada. A ella se le pasa como siempre el número de la unidad (80 o 81h) en el registro DL. Si la unidad correspondiente no está presente, después de la llamada de función devuelve el valor 0 en el registro AH.

Si allí se encuentran el valor 1 0 2, el dispositivo es una unidad de disquetes. El valor 3 finalmente, indica que hay un disco duro conectado. En este caso, los registros CX y DX contienen el número de sectores en este disco duro. Los dos registros forman un número de 32 bits, donde el registro CX contiene los bits altos, y el registro DX los bits bajos.

Discos duros y sus controladoras

Los avances en la tecnología de discos duros se pueden definir en particular con cuatro nombres, que describen cuatro familias de controladoras que se emplean en el ámbito del PC: ST506, ESDI, SCSI e IDE. Ya que de la controladora no solo depende el formato en que se guardan los datos en el disco duro, sino la tasa de transferencia entre el ordenador y el disco duro. Finalmente la controladora es la responsable de la capacidad de un disco duro, y de su velocidad, ya que un disco duro más rápido no sirve nada, cuando la controladora no aguanta la velocidad.

Las tasas de transferencia de datos máximas, que se pueden alcanzar con las diferentes controladoras se muestran en la siguiente tabla. Sin embargo hemos de añadir, que esto son valores máximos teóricos, que se alcanzan con poca frecuencia y que en la realidad dependen de toda una serie de otros factores. Ya que en el camino de las informaciones del disco duro a la pantalla (por ejemplo al cargar un documento de texto) se ha de pasar por, aparte de la controladora y el BIOS, por la aplicación en sí, y posiblemente algunos programas TSR.

Al fin y al cabo está el bus, que limita la velocidad de las controladoras de disco duro, ya que sigue trabajando a 8 MHz, aunque las CPU ya han desaparecido en los entornos de hasta 66 Mhz. Los valores indicados se reducen rápidamente, en algún caso concreto de carga de archivos, a menos de una quinta parte.

En los siguientes apartados se presentan las diferentes controladoras de disco duro, se analiza su estructura y se iluminan las ventajas y desventajas. También se habla del papel del BIOS durante la programación de estas controladoras.

Tabla : Tasas de transferencia máxima de datos de las diferentes controladoras de PC

Controladora	Tasa de transferencia máxima
ST506	1 MB por segundo
ESDI	2,5 MB por segundo
IDE	4 MB por segundo
SCSI	5 MB por segundo

Controladora ST506

Los primeros discos duros que encontraron aplicación en el ámbito del PC, se desarrollaron para el trabajo con las controladoras ST506 o compatibles. Como su nombre permite suponer, este estándar de controladora procede de la casa Seagate, que aún hoy juega un papel importante como fabricante de discos duros en el negocio de los PC.

Aún hoy las controladoras ST506 representan a una de las controladoras más difundidas, aunque los nuevos discos duros se suministran habitualmente con controladoras IDE. En la lista de precios de un suministrador de hardware puede ver discos duros que están pensados para una controladora ST506, y que se pueden reconocer en la denominación MFM/RLL. Con ello se denominan los dos métodos de grabación bajo los cuales la controladora guarda los datos en el disco duro. El formato que elija la controladora, habitualmente se puede determinar mediante interruptores DIP, donde se ha de preferir el método RLL, ya que garantiza una capacidad mayor en el disco duro.

La controladora ST506 ha fijado varios estándares en referencia al control del hardware gracias a su gran difusión, a lo que también ha aportado la predisposición unilateral del BIOS a este tipo de controladora. Las consecuencias se pueden notar aún hoy, y se han

encargado por ejemplo, de que las controladoras IDE y ESDI sean compatibles en diferentes aspectos con las controladoras ST506 (y deben serlo), pero después más sobre ello.

El hardware

Disco duro y controladora representan habitualmente dos componentes separados en el estándar ST506. Ya que la controladora se encuentra en una placa aparte y se encuentra «aparcada» en una de las ranuras de expansión del PC.

Una de estas controladoras habitualmente puede gestionar dos discos duros, donde normalmente se encuentran las llamadas «controladoras combi», y que además pueden acoger «bajo sus alas» dos unidades de disquetes. La controladora se conecta con el disco duro mediante dos cables: la línea de datos de 20 polos y la línea de control de 34 polos.

Al contrario que el cable de datos, que para cada uno de los discos duros existe por separado, los discos comparten el cable de control, que por ello posee dos extremos. A través del cable de control se controla la posición de los cabezales de lectura/escritura, así como se pone en marcha la lectura o escritura de sectores. Mediante el cable de datos se intercambian los datos a leer o escribir entre la controladora y el disco duro, donde también las informaciones de control pasan por el cable, y se guardan conjuntamente con el sector en el disco. La transmisión, sin embargo, no se realiza en binario, sino serie y analógica, con lo que es trabajo de la controladora convertir las oscilaciones de flujo en secuencias de bits.

La tasa de transferencia es fundamentalmente 5 Mbits/s, si el disco duro se formateó con el formato MFM, y 7,5 Mbits/s si se emplea el formato RLL. Pero como de este flujo de datos hay que filtrar las informaciones de control, la llamada «tasa de datos útil» está claramente por debajo. Pero aun así es medio MByte el que se puede procesar por segundo con el método MFM, mientras que el formato RLL llega incluso a 0,75 MByte. Pero como siempre en el entorno de los discos duros, esto son sólo valores teóricos que no tienen en cuenta los tiempos de conmutación entre cilindros y pistas, y que además presuponen que se puede leer sector a sector. Pero esto no siempre es el caso, como muestra el capítulo sobre el factor de intercalado.

La tasa de transferencia mayor del método RLL resulta del método de grabación más eficiente, que se nota en la cantidad de sectores, que se guardan en cada pista. Si en el método MFM por norma son sólo 17 sectores, el método RLL consigue casi diez sectores más, 26. La velocidad de giro del disco en ambos formatos es de 3600 revoluciones por minuto.

Ya que en los tiempos, en los que IBM comercializó su XT y sólo había disponibles controladoras ST506, se afinaron las funciones de los nuevos discos duros de la ROM BIOS a este tipo de controladora, y con ello se limitaron las posibilidades de rendimiento de aquel entonces. Así se limitó la cantidad de unidades de disco a dos, el número máximo de cilindros a 1024, el de sectores por pista a 63 y el de cabezales a 16. Además de ello se fijó el tamaño de sector a 512 Byte, lo que en combinación con los otros factores da una capacidad máxima de 504 MByte, más no es posible.

Que ya existen discos duros con capacidades mayores para el disco duro simplemente se debe a que se le hace creer al sistema en la existencia de dos discos, pero que internamente se proyectan sobre uno solo. Con ello se pueden conseguir capacidades hasta un Gigabyte, aunque estos discos no se pueden obtener para conectarlas a controladoras ST506. Ya que el que trabaja con semejante disco, espera también velocidades de acceso mayores, que no se pueden obtener con una controladora ST506.

Por ello, este tipo de controladora desaparecerá cada vez más del mercado en los siguientes años.

Controladoras ESDI

Después de la controladora ST506 vino primero ESDI, el «Enhanced Small Devices Interfaz», a la que apuesta sobre todo IBM con sus modelos PS/2. En muchos aspectos representa un desarrollo del estándar ST506, sin que se haya cambiado por ejemplo el interfaz de programación. Por ello se pueden utilizar los discos ESDI sin problemas con ordenadores en los que el BIOS fue «programado» simplemente para controladoras ST506.

A diferencia de los ST506, no se transmite, como en ESDI, cada cambio de flujo individual a través de la línea de datos vía serie a la controladora. En vez de eso, una parte de la lógica de decodificación, el llamado separador de datos, ya se encuentra en el disco duro. Prepara las informaciones leídas, y sólo transmite datos útiles en forma digital a la controladora en sí.

Ya que la controladora y el separador de datos trabajan de forma paralela, la tasa de transferencia se pudo elevar hasta 10 Mbits/s. Comparado con un disco MFM ST506 esto sólo es el doble, pero se añade una ventaja importante. Las controladoras ESDI generalmente están equipadas, al contrario que las controladoras ST506, con un buffer de sectores, que con ello hace posible un factor de intercalado (interleave) de 1:1. Mientras que un disco ST506 necesita, a causa del factor de intercalado 1:3 o hasta 1:6, un mínimo de tres sino hasta seis revoluciones para transferir una pista completa, el disco ESDI lo consigue en una sola. La velocidad de acceso se vuelve a aumentar en un factor de tres hasta seis.

Además, el estándar ESDI prevé, aparte de la tasa de transferencia utilizada comúnmente de 10 Mbits/s, tasas de transferencia de 15, 20 y hasta 24 Mbits/s. Estas controladoras son muy raras, y habitualmente muy caras, de modo que la mayoría de controladoras ESDI trabajan con 10 Mbits/s.

Por cierto, el separador de datos no es la única pieza de «inteligencia» que se monta en los discos duros ESDI al contrario que en los discos ST506, ya que un disco ESDI guarda, además de todo, informaciones sobre su formato físico y las direcciones de sectores defectuosos, y suministra estas informaciones a la controladora bajo petición. Esta se puede adaptar con ello al disco duro conectado, lo que en discos ST506 ha de garantizar el usuario, seleccionando un formato de datos correspondiente en el programa de configuración del BIOS.

El asunto con el BIOS

Esta información, que se guarda después en la RAM CMOS del reloj mantenido por baterías del AT, también necesita un ordenador con controladora ESDI. Ya que en todos los casos el BIOS ha de conocer las «medidas» del disco duro y enviar esto al controlador de dispositivos del DOS.

Pero como la controladora ESDI puede obtener los datos límite del disco duro, estas informaciones en el BIOS no han de coincidir con el formato real del disco. Y esta discrepancia a veces ni siquiera se puede evitar, ya que cada BIOS solo conoce un número determinado de discos duros y sus medidas.

En el caso de un disco duro ST506 se tiene un gran problema, si el disco duro que hay montado no se puede encontrar en esta lista. Ya que entonces se ha de seleccionar un disco, cuyo formato se acerque al máximo al del propio, y esto frecuentemente quiere decir:

renunciar a algunas pistas o incluso cabezales, ya que siempre se ha de elegir un formato inferior, y nunca superior. Ya que de lo contrario, el BIOS accederá después a sectores que no existen, y eso nos traerá problemas seguros.

La cosa se complica sobremanera, cuando el BIOS no ofrece ningún tipo de disco duro, que trabaje con el mismo número de sectores por pista. Ya que entonces se ha de seleccionar el siguiente tamaño de sector inferior, y eso significa: en cada pista del disco se regalan una serie de sectores, potenciando con ello el número de sectores sin utilizar hasta límites inaceptables.

Esto es algo que no se ha de temer en los discos ST506, ya que por norma trabajan con 17 o 26 sectores según el método de grabación. Pero en el caso de los discos ESDI, la cosa tiene otro aspecto. Ya que aquí los formatos normales son de 34 o 36 sectores por pista. Si el BIOS sólo conoce discos con 26 sectores, se habría de dejar aproximadamente un tercio del caro disco duro sin emplear.

Pero esto no es necesario, ya que la controladora puede obtener las verdaderas medidas de un disco duro y ponerlas en relación con los valores que hay guardados en el BIOS. Estos se le comunican automáticamente durante su inicialización por parte del BIOS. La controladora se pone con ello en disposición de proyectar el formato indicado por el BIOS sobre el formato real, donde el BIOS ha de convertir las direcciones de sector indicadas en cada acceso. Esto cuesta un poco de tiempo (realmente sólo un poco), pero consigue que el disco duro se pueda hacer funcionar con prácticamente cualquier formato de BIOS, mientras no prevea una capacidad total mayor.

Las capacidades para realizar una proyección de este tipo, depende de la controladora ESDI. Algunas solo soportan formatos determinados, mientras que otras son flexibles y aceptan cualquier formato. Esto también ayuda, cuando los discos ESDI llega a los límites del BIOS, así por ejemplo discos con más de 1024 cilindros. En vez de dejar sin utilizar los cilindros adicionales, estos discos habitualmente simulan más sectores por pista de los que realmente existen, aunque el límite de 63 sectores por pista les deja poco margen de juego en este aspecto.

SCSI

El estándar SCSI (pronunciado: escasi) no sólo pone a disposición un interfaz dedicado para el disco duro, sino para poder conectar hasta ocho dispositivos totalmente diferentes. Ya que aparte de discos duros, también se ofrecen Streamer, unidades CD-ROM o Scanner, que están preparados para este interfaz.

A diferencia de los estándares de controladora actuales, el «Small Computer System Interfaz» no solo se encuentra en el ámbito del PC, sino también en muchos sistemas 68000 (Mac, Atari ST) y Workstations. Su gracia está en que desacoplan los dispositivos conectados del ordenador en sí, ya que los dispositivos se comunican con la controladora mediante un bus de sistema propio, que se separa del Bus del PC.

Ya no solo la ocupación de líneas del Bus SCSI está estandarizada, sino también los comandos que se envían a través de él a los dispositivos, los dispositivos SCSI se pueden intercambiar libremente entre diferentes sistemas, y sólo la controladora SCSI en sí ha de estar adaptada al sistema en que se encuentra. Pero en la práctica, esta compatibilidad no siempre es lo que parece, especialmente en el caso de controladoras SCSI baratas, que hoy en día se pueden conseguir por unos cuantos miles de pesetas. Como estándar se definen las controladoras SCSI de Adaptec, pero también Western Digital y Future Domain representan a los «major players» en este segmento de mercado.

El bus de sistema que une los diferentes dispositivos SCSI entre sí y que habitualmente sale al exterior en forma de un conector de 80 polos, está organizado de forma paralela, y permite la transmisión simultáneamente de 8 bits. Las nuevas versiones del estándar SCSI, SCSI II, también permite buses de 16 bits, aunque hasta ahora no ha tenido una gran difusión en el ámbito del PC.

El cuento de las tasas de transmisión

A los distribuidores de controladoras SCSI les gusta atraer a sus clientes con tasas de transmisión de datos de 4 o 5 MByte/s, que en los oídos de algún usuario torturado del DOS debe sonar como música celestial. Pero las esperanzas despertadas, en la práctica rápidamente decrecen. Ya que la controladora SCSI seguramente obtiene esta tasa de transmisión, pero el disco duro conectado seguramente no. Se pueden esperar entre 1,5 y 2 MByte/s, si no se invierte en una controladora EISA cara, que consigue, gracias a las mejoras de este sistema de BUS, hasta 2,5 MByte/s.

Los discos SCSI siguen con la moda que comenzó con los discos ESDI, e integran gran parte de la electrónica de control necesaria directamente en el disco duro. En realidad tampoco es posible de otra forma en este sistema, ya que la controladora se quiere mantener independiente del dispositivo, y por ello no debe preocuparse mucho de las características y particularidades de un disco duro.

La gran cercanía entre el disco duro y su electrónica de control ofrece algunas ventajas, no sólo porque los caminos son mas cortos. Ya que también se hace más sencillo el simularle un formato de disco determinado a la controladora, que en realidad no existe. No en vano la controladora SCSI, al igual que las controladoras ESDI, pregunta el formato de los dispositivos conectados durante el arranque del sistema, y lleva estas informaciones al sistema.

A ello le ayuda el BIOS que tiene integrada, ya que los sistemas SCSI dependen de un BIOS propia, ya que en la parte de software no soportan el estándar ST506, que parte de la ROM BIOS. Las funciones originales de la interrupción de disco del BIOS se sustituyen por las del BIOS SCSI. Por desgracia este BIOS no sirve para el Protected Mode, en el que los «sistemas operativos» como Novell u OS/2 han de acceder directamente al disco duro, y frecuentemente sólo soportan el estándar ST506. En ese caso se necesitan los controladores de dispositivos adecuados, pero precisamente estos son difíciles de conseguir. Aquí seguramente está uno de los inconvenientes más grandes del interfaz SCSI.

Por otra parte, si se dispone del controlador correspondiente, la ampliación de la capacidad del disco no es ningún problema, al contrario que en otros sistemas. Ya que para ello sólo es necesario montar un disco duro adicional, y unirlo con el bus SCSI. Del resto se encarga la controladora SCSI en conjunto con los controladores correspondientes.

IDE

La estrella entre las controladoras de disco duro es sin duda el interfaz IDE (Intelligent Drive Electronics), que hoy se puede encontrar en casi cualquier PC nuevo. El desarrollo de este estándar se remonta sólo al año 1984, cuando el fabricante de PC Compaq le encargó

al especialista de discos duros Western Digital el desarrollo de una controladora compatible ST506, que, a causa de la falta de espacio, tuviera la electrónica de control directamente en el disco duro.

Así se ha quedado hasta hoy en día, y en realidad, la unión de la controladora a la unidad de disco es la característica más destacable de los discos IDE. Así que el que compre un disco IDE, ya está comprando directamente la controladora. Esta es también la razón, por la cual en los discos IDE no se necesitan líneas de datos y de control separadas, sino simplemente un cable de 40 polos, que se une directamente al bus del sistema.

A esta circunstancia, los discos duros le deben su segundo nombre, ya que frecuentemente se denominan como «discos de Bus AT». Pero esto no debe confundirle, pensando que estos sistemas están limitados a los AT y sus buses de datos de 16 bits de anchura. Ya que también existen, de forma análoga, discos de Bus XT, que se apoyan en el Bus de 8 bits del XT. Pero como los XT ya no tienen una difusión significativa, este hecho es menos conocido.

En realidad los discos IDE se pueden montar directamente en la placa madre, pero habitualmente, este tipo de discos se suministran con una pequeña tarjeta de ampliación, que se ha de montar en una ranura de expansión, y que apenas representa una conexión entre el Bus del sistema y el cable IDE.

A causa de la estrecha unión entre el disco y la controladora, IDE disfruta de algunas ventajas, que tienen las controladoras SCSI. A ello se suma la capacidad de emular formatos cualesquiera, así como el Multiple Zone Recording, que crea más espacio en el disco o el caché de pistas, con cuya ayuda se pueden leer pistas completas que se mantienen en un buffer de caché interno. Por ello los discos IDE se pueden hacer funcionar con un factor de intercalado de 1:1, lo que garantiza un acceso rápido y ayuda a que los discos IDE no se queden atrás en cuanto a rendimiento, comparándolos con los sistemas de disco ESDI. IDE reúne por ello las ventajas de los otros tres estándares de controladora, ya que es flexible como SCSI, rápido como ESDI y gracias a su compatibilidad con el estándar ST506 se puede montar en la mayoría de los sistemas PC.

Además de ello, los discos IDE se caracterizan, en comparación con otros conjuntos de disco-controladora, por un consumo muy bajo, lo que la hace ideal para ser utilizada en ordenadores del tipo Laptop y Notebook.

Especialmente para el trabajo con estos sistemas, muchos discos IDE reconocen comandos especiales, que los conmutan a un estado de «hibernación», con lo que se reduce aún más el consumo de corriente. Estos comandos normalmente sólo se pueden utilizar en conexión con controladores especiales o con un BIOS, que esté diseñada especialmente para el trabajo conjunto con discos IDE. Ya que desde el punto de vista del BIOS, los discos IDE se comportan como controladoras normales ST-506, lo que hace mucho más sencilla la integración de estos discos en sistemas existentes.

Hoy en día se está definiendo un estándar único para discos IDE, que en un futuro hará posible un soporte directo de discos IDE con muchos sistemas de BIOS. Entonces se podrán utilizar mejor las posibilidades ampliadas de estos discos, que hoy por hoy no se utilizan en muchos de los sistemas.

De la controladora a la memoria

Independientemente de la velocidad del disco duro y de la controladora, la forma en que se transfieren los datos de la controladora a la memoria, decide la auténtica velocidad del conjunto controladora-disco duro. Se pueden utilizar cuatro métodos diferentes:

Programmed I/O

En el Programmed I/O, la transferencia de datos entre controladora y la memoria principal se desarrolla a través de los diferentes ports I/O (E/S) de la controladora, que también sirven para la transmisión de comandos. En el lado del software se encuentra el programa correspondiente con los comandos de lenguaje máquina: IN y OUT. Pero esto significa que cada byte o word individual se ha de pasar por la CPU.

La tasa de transferencia de datos no sólo se limita por los valores límite del Bus PC, sino también por el rendimiento de la CPU. A pesar de que el Bus ISA permita una tasa de transferencia de datos máxima de 5,33 MByte/s en la transferencia de words, esta no se puede alcanzar del todo, aún con una CPU rápida. A pesar de todo, los 386 y los 486 pueden servir bastante bien la controladora, ya que recogen más datos de los que puede suministrar la misma. Ya que incluso 3 o 4 MByte/s sólo se pueden conseguir con discos duros extremadamente rápidos y caros.

Memory-Mapped I/O

La CPU puede recoger los datos de la controladora de forma más rápida, si los deja en una zona de memoria fija, donde habitualmente se elige un segmento por encima de la RAM de vídeo. Ya que entonces se puede realizar la transferencia de los datos a una zona de memoria del programa correspondiente con ayuda de instrucciones MOV, que trabajan más rápidamente que los accesos mediante IN y OUT.

Con ayuda de esta técnica se pueden obtener más rápidamente los datos en las CPU modernas, que los puede suministrar las controladoras. Ya que el valor teórico máximo de 8 MByte/s, la controladora no lo alcanza nunca, e incluso los valores más realistas de 5 a 6 MByte/s no se pueden conseguir.

DMA

Más conocido aún que los dos procedimientos nombrados hasta ahora, es la transferencia DMA, que en el PC se soporta con un chip DMA propio. Debe posibilitar la transferencia de datos de un dispositivo (disco duro, disquete, CD-ROM, etc.) a la memoria y evitar con ello el largo camino a través de la CPU (DMA: Direct Memory Access). La idea es buena, ya que el software sólo ha de indicarle al controlador DMA cuantos Byte se han de transportar de donde adónde, pero en el PC se realizó de una manera bastante burda.

Ya que el controlador DMA que se emplea en el PC no solo es bastante inflexible (lo que aún se le podría perdonar), sino sobre todo lento. Tan lento que el acceso según el procedimiento Programmed I/O en el peor de los casos es más rápido a partir de los 386 y 486. Ya que como anacronismo en la historia de los PC, el controlador DMA en los AT y sus sucesores sigo haciéndose funcionar a 4 MHz, donde en los PC ya alcanzaba los 4,77 MHz. Allí la transferencia DMA es más rápida que el Programmed I/O, pero sólo allí. De esta forma no se puede pasar de más de 2 MByte/s.

Por ello, la mayoría de los discos duros modernos ya no se controlan según el método DMA, aunque muchos de ellos soporten este método aparte del de Programmed I/O.

Busmaster-DMA

Otra forma del Direct Memory Access es el Busmaster DMA, pero que no tiene nada que ver con el chip de DMA que hay integrado en la placa del ordenador. En este procedimiento, la controladora del disco duro desconecta la CPU del Bus, y transfiere los datos con ayuda de un controlador Busmaster DMA con control propio. Con ello se pueden conseguir tasas de transferencia de hasta 8 MByte/s, aunque esta capacidad aumenta el precio de la controladora en proporción. Busmaster DMA sólo se emplea en el caso de controladoras SCSI especialmente capaces.

Goodbye Protected-Mode

Ambas formas de la transferencia DMA sólo se pueden utilizar en el mundo del modo Real (Real-Mode), y llevan al procesador a un «cuelgue» si no se toman las medidas contrarias adecuadas, cuando trabaja en el Protected Mode o en el Virtual-86 Mode. El problema es la gestión de memoria virtual, que se representa por la Memory-Management-Unit (MMU) en la CPU. Forma direcciones virtuales con las que trabaja un programa en este modo, y las proyecta sobre las direcciones verdaderas (físicas) en la memoria.

Pero de esto el programa no se entera, ya que nunca entra en contacto con las direcciones físicas, y también el controlador DMA no queda afectado de ello, ya que no tiene acceso a la MMU de la CPU. Como consecuencia, los datos se desplazan a zonas totalmente desconocidas, lo que a la larga sólo tiene como consecuencia un error del ordenador.

Y este destino no sólo alcanza a los programas cuando Windows trabaja en el Protected Mode, o cuando se utiliza un sistema operativo como OS/2. Ya que incluso bajo DOS el procesador se conmuta al modo Virtual-86, en cuanto se instala el controlador de dispositivos EMM386.EXE, que sirve para la emulación de memoria expandida y que depende de la gestión de memoria virtual del procesador.

Sólo hay un modo de evitarlo, y es la vigilancia del controlador DMA, como se hace posible en el Protected Mode mediante el control de todos los ports I/O. Así, por ejemplo Windows instala un Control-Monitor virtual en el fondo, que vigila la programación del controlador DMA mediante el BIOS o un programa, y que convierte las direcciones virtuales indicadas en las direcciones físicas verdaderas, antes de que se escriban en los registros del controlador DMA.

Grabación de las informaciones en un disco duro

Si se pregunta la forma en que las informaciones se guardan en la superficie del disco duro, primero se ha de olvidar la representación de una codificación binaria. Ya que no se pueden guardar unos y ceros con ayuda de partículas magnetizables, ya que en ese caso se habrían de reflejar los dos estados «magnetizado» y «no magnetizado» para los valores 0 y 1. Pero precisamente esto no es posible, como mostraremos a continuación.

Ya que si se siguiera esta propuesta, la secuencia de ceros y unos aparecería como secuencia de partículas magnetizadas y no magnetizadas. Pero con ello, el cabezal de lectura del disco duro no podría distinguir las diferentes partículas magnéticas de esta secuencia. Así que no sabría si está tratando con tres o con cinco ceros.

En esta situación sólo se podría conseguir ayuda, si el cabezal conociera la «longitud» de una partícula magnética, pudiendo poner entonces el tiempo transcurrido entre una partícula y otra en función de la cantidad de partículas magnéticas y con ello en relación con los bits. Así que se necesita una especie de pulso de reloj, que con cada impulso indique que ha llegado el tiempo de un nuevo bit.

Pero el tiempo constante, que se necesita para la creación de un pulso, no se puede determinar con exactitud a causa de diferentes factores. La culpa la tiene por ejemplo la velocidad de rotación de disco duro, que dentro de ciertos límites siempre varía, así como el hecho de que nunca se puede magnetizar una única partícula magnética, sino siempre todo un grupo, cuyo número no siempre es constante.

Lo que sí se pueden grabar sin problemas, son los llamados cambios de flujo, transiciones cortas entre partículas magnetizadas y no magnetizadas. Crean en el cabezal de lectura del disco duro un impulso eléctrico que se conduce a la electrónica y que allí sirve para la decodificación de informaciones guardadas en forma de unos y ceros.

Y precisamente esta codificación y decodificación de informaciones binarias es la que desde siempre ha ocupado a los desarrolladores del hardware. Ya que la cantidad de cambios de flujo que se pueden grabar en un disco duro por milímetro cuadrado, está limitado. A pesar de que depende de la estructura del material magnetizable, de la altura de vuelo del cabezal de lectura/escritura y de su sensibilidad, pero en cualquier caso representa una magnitud limitada. Quien consiga, con el mismo número de cambios de flujo, colocar más unos y ceros en el disco, siempre estará delante en la carrera de las capacidades de disco cada vez mayores.

El procedimiento FM

La codificación más sencilla de ceros y unos con ayuda de cambios de flujo consiste en la grabación de un cambio de flujo para cada uno, y el omitir el cambio de flujo para cada cero. Con ello de nuevo se repiten los problemas de los que ya hemos hablado anteriormente. ya que también aquí se necesita un pulso, para que la ausencia de cambios de flujo durante un tiempo determinado se pueda reconocer con una secuencia de ceros.

En el método más sencillo y antiguo de grabación, el de la llamada modulación de frecuencia (FM: frequency modulation), la señal de pulso se graba directamente en el disco duro en forma de un cambio de flujo. En una secuencia temporal constante se graban en el disco duro cambios de flujo, a los que se sincroniza la electrónica del disco duro, para poder leer los bits de datos, ya que estos se encuentran entre los cambios de flujo de pulso. Un cambio de flujo para un uno, y ningún cambio de flujo para un cero.

Este procedimiento se puede hacer con una electrónica de control relativamente simple, pero tiene un gran inconveniente: de esta forma, cada bit de datos consume dos cambios de flujo, y reduce con ello la posible capacidad del disco duro a la mitad.

El procedimiento MFM

El procedimiento MFM acaba con este «despilfarro» de cambios de flujo, y es el que se emplea en la mayoría de discos duros hoy en día. Representa una modificación del procedimiento FM, lo que le ha traído el nombre de «modified frequency modulation».

En este procedimiento los cambios de flujo de pulso se utilizan incorrectamente como señales de datos, desplazándolos con respecto al pulso normal. Pero para comenzar, todo se queda como está: ya que un cero se compone de un cambio de flujo como siempre, al que no sigue ningún cambio de flujo hasta el siguiente cambio de flujo de pulso. Pero un uno sólo se graba como un cambio de flujo, y concretamente allí, donde también se puede encontrar el cambio de flujo en el método FM. Pero el cambio de flujo precedente falta. De esta forma se pueden grabar ceros y unos con ayuda de un único cambio de flujo. Secuencias más largas de ceros y unos se representan como una fila continuada de cambios de flujo, que sin embargo se encuentran desfasados en su posición.

Es condición para este método de codificación es una técnica de grabación y una electrónica de control mejoradas, para que el disco duro se pueda sincronizar a la secuencia normal de cambios de flujo, durante una secuencia larga de ceros. Si después de la lectura de un cero transcurre más tiempo que el usual, hasta que aparece el siguiente cambio de flujo, se puede reconocer con ello un uno. Si el siguiente cambio de flujo aparece después del tiempo acostumbrado, es de nuevo un uno.

Sólo aparecen problemas cuando después de un uno aparece un cero, que de nuevo requiere un cambio de flujo en la posición de pulso normal. Ya que el tiempo entre el cambio de flujo del uno y del cero es en ese caso tan sólo la mitad de la distancia normal entre dos cambios de flujo. Pero esto no se puede hacer, ya que no se puede reducir la distancia entre dos cambios de flujo, ya que el cabezal y la electrónica ya no son capaces de seguirlo. Por eso, para un cero que viene después de un uno, no se guarda ningún cambio de flujo.

Así que el siguiente cambio de flujo aparece después de una vez y media de tiempo para un cambio de flujo (combinación de bits 100b) o incluso después del doble de este tiempo (combinación de bits 101b). Esto lo ilustra la siguiente figura.

El procedimiento RLL

Quien piense que la grabación de datos haya llegado al límite con el método MFM, está equivocado. Ya que con el método RLL hace tiempo que existe un método de codificación, que de nuevo consigue colocar hasta un 50 por ciento extra de informaciones en el disco, comparado con el método MFM.

En este procedimiento, los unos se graban como cambios de flujo, y los ceros como la ausencia de cambios de flujo, sin que se grabe adicionalmente un cambio de flujo de pulso. La electrónica del disco duro ha de contar el pulso por ella misma. Pero esto sólo es posible con unas revoluciones muy constantes y una mejora del cabezal de lectura y su electrónica, y si el número de ceros entre los unos no es demasiado grande, ya que con cada cero de esta cadena, aumenta el tiempo hasta el siguiente cambio de flujo, y con el la probabilidad de que la electrónica del disco duro pierda el ritmo.

Por otra parte, los unos no pueden seguir con demasiada frecuencia, ya que de lo contrario la electrónica amontona los cambios de flujo, y ya no puede seguirlos.

Para cumplir con ambas condiciones la electrónica del disco duro proyecta las secuencias de bits en el procedimiento RLL (RLL: Run Length Limited) a otro código, que es el que se guarda. El nuevo código es el doble de largo que el código original, pero se encarga de que la serie de ceros no sea demasiado larga, y de que las distancias entre unos no se hagan demasiado cortas. Condición primordial es naturalmente en cualquier caso, que las informaciones guardadas se pueden decodificar sin problemas en las secuencias de bits originales, sin que aparezcan ambigüedades.

Naturalmente se puede pensar en infinitos patrones de código, según lo grande que sea el número máximo de ceros consecutivos y lo pequeño que sea su número mínimo. Se utilizan dos métodos, que según estos dos tamaños se denominan como RLL 2,7 y RLL 3,9.

RLL 2,7 es el método estándar, que se emplea en la mayoría de discos duros modernos, y que han de ser lo más pequeños posible, con la mayor capacidad posible. Entre dos unos aparecen en ese caso un mínimo de dos y un máximo de siete ceros. Al contrario que en el método MFM se obtiene un aumento de la capacidad del 50 por ciento.

De forma aún más densa se pueden empaquetar las informaciones con el formato RLL 3,9 que también se denomina como «advanced RLL». Entre dos unos se encuentran un mínimo de 3 ceros, y un máximo de 9 ceros.

La siguiente tabla muestra la conversión de código en RLL 2,7. A lo mejor se da cuenta de que algunos Byte, como por ejemplo 00000001b no se pueden transformar a primera vista. Y no se puede olvidar que a este nivel se trabaja fundamentalmente con sectores y no con Byte individuales. Por ello, también un byte como 00000001b se puede transformar, utilizando los bits de los siguientes Byte en la codificación.

Tabla : Tabla de código para el procedimiento RLL 2,7

Patrón de bits a guardar	Código
000	000100
10	0100
010	100100
0010	00100100
11	1000
011	001000
001	00001000

Sólo es problemática la codificación del último byte de un sector, ya que entonces el código debe funcionar, puesto que no hay Byte subsiguientes. Pero en la práctica este problema se resuelve de la siguiente manera: la controladora añade un byte adecuado. Los bits superfluos se vuelven a cortar después de la lectura, y con ello el último byte del sector se devuelve correctamente.

Más pequeño, más rápido, más barato

«Smaller, Faster, Cheaper», apenas ningún otro desarrollo en el ámbito del PC parece haber estado bajo este lema mágico en los últimos años de la industria microelectrónica, como el desarrollo en la técnica de discos duros. He aquí unos cuantos factores para aclarar esto:

*Los tiempos de acceso han bajado de treinta milisegundos a unos buenos diez milisegundos en los modelos más rápidos.

*Las capacidades normales han subido hasta varios cientos de MB. Los discos duros con una capacidad de más de 1 GB ya no son rarezas.

*Un MegaByte de capacidad de disco duro hoy en día cuesta una media de unas 700 pesetas.

Así que la capacidad de disco duro no solo es como unas diez veces más barata que la memoria RAM, sino que encima tiene ventaja en cuanto a densidad de información, comparándola con los Mega-Chips. Pero esta ventaja se convierte rápidamente en humo, si se consideran los tiempos de acceso, que están separados por varias potencias de diez.

Los discos duros han crecido en todas sus dimensiones sobre todo por que se ha optimizado cada uno de los procesos de fabricación y cada uno de los detalles de la técnica de grabación y reproducción. Esto ya comienza con el ordenamiento de los sectores individuales en el disco y el llamado factor de intercalado.

El factor de intercalado (Interleave)

Hoy en día las controladoras de disco duro se han convertido en tan rápidas, que se pueden permitir en un acceso de lectura a un sector, leer directamente toda la pista, aunque el programa no haya pedido los demás sectores. Pero esto ocurre en la mayoría de los casos, si el disco no está muy fragmentado. Y en este caso la controladora del disco duro ya no ha de leer, decodificar y preparar los datos del disco duro en uno de los subsiguientes accesos, sino que los puede obtener directamente de su buffer interno. Esto naturalmente acelera enormemente el acceso.

Pero realmente, este sólo es el caso en los tipos más nuevos de controladoras (SCSI, ESDI e IDE), ya las controladoras ST506 sólo poseen un buffer de sectores interno, en el que naturalmente no se puede guardar un sector completo. Pero aún peor: ya que el sector leído sólo se puede pasar a la CPU cuando se ha leído el último byte, el buffer de sectores se sigue utilizando un instante, mientras el siguiente sector está pasando por debajo del cabezal, y por ello este no se puede leer de forma preventiva.

Si el siguiente acceso de lectura realmente se hace en este sector, no se puede leer inmediatamente, ya que transcurre casi una vuelta completa del disco duro, hasta que vuelve a aparecer debajo del cabezal. Y como este proceso se repite con cada sector, y baja sensiblemente la velocidad de acceso al disco, los desarrolladores del software han buscado desde el principio una solución a este problema. Y la encontraron en el llamado «Interleaving», el desplazamiento de sectores lógicos con respecto a los físicos.

Con ello se consigue, que después de un sector lógico pasen unos cuantos sectores adicionales por delante del cabezal, hasta que aparece el siguiente sector lógico. Y concretamente tantos sectores, para que el siguiente sector aparezca justo cuando el antecesor haya sido suministrado y el BIOS haya dado la orden de buscar el siguiente sector. Sólo en ese caso el desplazamiento es el óptimo.

El interleaving se mide en el llamado factor de Interleave, que es mucho más conocido que todo el proceso en sí. Designa la cantidad de sectores, en la cual se han desplazado los números de sector lógico con respecto a los números verdaderos (físicos) de sector. En los discos duros originales que IBM montaba en su XT, este factor era 1:6, y después en el AT, 1:3. Sin embargo, el disco duro del AT se podía bajar al factor 1:2, con lo que se subía la velocidad de acceso.

Hoy en día, los factores de Interleave de 1:1 son habituales, lo que no significa otra cosa, que ya no se utiliza en interleaving.

En el caso de un Interleave de 1:6 aún es necesario saltarse cinco sectores, antes de que llegue el siguiente sector lógico, y en el caso de 1:3 al fin y al cabo son dos sectores. La siguiente figura le muestra el ordenamiento desplazado de sectores en el ejemplo de una pista con 17 sectores.

Tabla: Interleaving de los primeros discos duros XT y AT de IBM

AT		XT	
Sectores físicos	Sectores lógicos	Sectores físicos	Sectores lógicos
1	1	1	1
2	7	2	4
3	13	3	7
4	2	4	10
5	8	5	13
6	14	6	16
7	3	7	2
8	9	8	5
9	15	9	8
10	4	10	11
11	10	11	14
12	16	12	17
13	5	13	3
14	11	14	6
15	17	15	9
16	6	16	12
17	12	17	15

A pesar de las ventajas indiscutibles de este procedimiento, no se puede olvidar, que en principio es deseable no tener Interleaving, ya que sólo entonces se puede leer una pista completa durante una revolución del disco duro. De lo contrario se necesitan, según el factor de Interleave, dos, tres, o más vueltas.

Instalación del Interleave

El Interleave se instala durante el llamado «Low-Level-Format» del disco duro, durante el cual se graban las marcas de direcciones y los números de sector en el disco duro aún virgen. Independientemente del sistema operativo, que luego se graba en el disco duro en un proceso de formateo separado, aquí se ponen las bases para el reconocimiento de los sectores.

Ya que el número lógico del sector individual se selecciona por el programa de formateo de bajo nivel (Low Level), es posible un desplazamiento de los sectores. Por ello, la mayoría de los programas de este tipo preguntan el factor de intercalado deseado. Si se selecciona el valor equivocado, se pueden tener desagradables sorpresas después, ya que el propio ordenador puede llegar a ser mucho más lento al cargar archivos o programas que el del amigo o compañero.

En ese caso se eligió el factor de intercalado erróneo, pero este problema se puede arreglar a posteriori. Con los programas de utilidades adecuados, como por ejemplo las Norton Utilities. Averiguan el factor de intercalado óptimo y realizan el formateo a bajo nivel a posteriori. Esto funciona realmente bien, y a veces sólo se tarda unos minutos, consiguiendo con ello un mejor rendimiento del disco duro.

Puede que tema por su disco duro, y que vea desaparecer sus datos, ya que estos programas hurgan profundamente en las interioridades del disco duro. Suena paradójico, pero precisamente por esto no hay motivos de preocuparse (siempre que se haya programado limpiamente). Ya que estos programas trabajan por debajo del nivel del DOS, DOS ni se entera de estas modificaciones. Al fin y al cabo no sabe de donde procede el sector que está escribiendo o leyendo, y mientras los datos se lean antes del nuevo formateo de la pista, devolviéndolos después a su lugar, para DOS no cambia absolutamente nada.

Track Skewing y Cylinder Skewing

Una codificación adecuada de los números de sector lógicos no solo sale a cuenta referido a la lectura secuencial de una pista. Ya que después de leer una pista, habitualmente sigue el acceso al siguiente cilindro. Pero de eso se encarga ya el sistema operativo, que en su numeración de sectores primero pasa por todos los diferentes cilindros de una pista, antes de cambiar al siguiente cilindro. Ya que el cambio de un cilindro a otro sólo necesita una breve conmutación del cabezal de lectura/escritura correspondiente, mientras que el acceso a otra pista implica un posicionamiento de brazo completo de lectura/escritura lo que consume mucho más tiempo.

A pesar de todo: incluso la conmutación del cabezal de lectura/escritura necesita algo de tiempo, y en ese tiempo el disco duro naturalmente sigue girando. Si se acaba de leer el último sector de un cilindro, el primer sector del siguiente cilindro ya ha pasado volando por debajo del cabezal de lectura/escritura, de modo que es necesaria casi una vuelta completa, antes de que pueda realizarse el acceso.

Para evitar esto, también entre las pistas de un cilindro se realiza una especie de interleaving, que se denomina como Cylinder Skewing. Los sectores en las diferentes pistas de un cilindro se desplazan de forma que, a pesar de conmutar al siguiente cabezal, se pueda leer inmediatamente el primer sector de la pista. Y también esto se ajusta durante el formateo de bajo nivel del disco, pero sin embargo no conocemos programas con los que se pueda realizar una optimización posterior, al contrario que en el caso del factor de intercalado.

Aparte del Cylinder Skewing, también existe el Track-Skewing. Trabaja según el mismo principio, pero tiene en cuenta el tiempo, que se necesita para mover el brazo completo de lectura/escritura hasta la siguiente pista.

Multiple Zone Recording

Si se quieren guardar más sectores en un disco duro, se puede utilizar un sencillo truco: en las pistas exteriores siempre hay más espacio que en las interiores, sencillamente por que su perímetro es mayor. Antes no se podía aprovechar esta circunstancia, ya que el número de sectores era constante y se determinaba por la capacidad de la pista más interior (más corta).

Con los modernos discos SCSI e IDE, que de todas formas simulan los datos identificativos en cuanto a número de cabezales, pistas y sectores, se ha hecho posible un formateo variable de las pistas individuales. Estos conjuntos de controladora y disco convierten el número de cabezal, cilindro y sector que envía el BIOS, a otra dirección de sector, con lo que se puede tener en cuenta la ampliación de las pistas exteriores.

Sin embargo, esto fuerza más la electrónica de la controladora y los cabezales de lectura/escritura, ya que en el mismo tiempo (de revolución) se han de leer o escribir más sectores en las pistas exteriores que en las interiores. Mientras que la longitud de los cambios de flujo disminuye, aumenta el número de datos que se pueden leer o escribir por revolución y con ello la tasa de datos.

Se puede aumentar la capacidad de un disco duro entre un 20 y un 50 por ciento con respecto al esquema rígido convencional.

Corrección de errores

Un criterio importante para la utilización de discos duros fue desde el principio su tolerancia de errores. Es imposible evitar impurezas en el material magnético, y esto provoca que haya algunos sectores del disco duro que no se puedan utilizar completamente.

Antes, los números de estos sectores defectuosos se imprimían en la carcasa del disco duro, ya que se descubrían (o debería ser así) durante el control de calidad del fabricante. Al llamar el programa de formateo de bajo nivel, el usuario tenía la obligación de introducir los números de los sectores defectuosos, para que se marcaran como tales, y posteriormente se pudieran reconocer por el sistema operativo. Esto tenía como consecuencia que estas zonas se marcaran en la FAT (File-Allocation-Table) como defectuosas, y que se excluyeran de la gestión normal de sectores.

En discos duros modernos, ante todo en discos IDE, estas listas de errores ya no se encuentran en la carcasa del disco duro. Ya que sus direcciones o bien se graban en pistas de datos separadas por el fabricante, o se reconocen durante el formateo a bajo nivel. Estos sectores simplemente se saltan, o se sustituyen por sectores que están en zonas protegidas. Allí también se guardan las tablas, que marcan los sectores defectuosos y sus sustituciones, de modo que la controladora del disco duro simplemente ha de cargar estas tablas durante su inicialización. Al acceder a un sector defectuoso, puede conmutar al sector que lo sustituye. Esto disminuye el acceso al disco duro, pero teniendo en cuenta que el porcentaje de sectores defectuosos, que es mínimo, prácticamente no tiene importancia.

Además de ello, hay muchos discos, que están en disposición de reconocer sectores defectuosos durante el funcionamiento. Sólo por eso se guarda con cada sector el llamado Error-Correction-Code, con ayuda del cual se pueden reconocer errores durante la grabación de un sector y en cierta medida también se pueden corregir. Estos sectores se marcan como defectuosos durante el funcionamiento, y se sustituyen por un sector de reserva, en el que se guardan las informaciones reconstruidas.

Pero esto sólo es posible en el caso de una estrecha unión entre disco duro y controladora, como únicamente la ofrecen los discos IDE.

Lo que ha de ir además

Además de las informaciones de bits en sí, es preciso colocar muchas mas cosas, en un disco duro. Ya que un simple sector necesita toda una serie de informaciones adicionales, para que pueda ser reconocido y leído correctamente por el disco duro. Este formato difiere entre las diferentes controladoras, pero en principio se guardan siempre las mismas informaciones, como por ejemplo el número de sector y cilindro, así como el Error-Correction-Code para el reconocimiento de errores. La siguiente figura muestra la estructura de un sector, como se puede encontrar en las controladoras ST506.

Puede ver, que cada sector se introduce con una serie de trece Byte de sincronización con el valor 00h (campo SYNC). Mediante estos patrones de bits también se crea una serie homogénea de cambios de flujo, que ayudan a la controladora a sincronizarse con el sector. A continuación viene la zona ID, mediante la cual el sector se identifica. Aquí se guarda ante todo el número de cilindro, cabezal y sector del sector (campos ZK/S). Estas informaciones se introducen mediante un byte especial de identificación (campo ID) y se cierran mediante el código de error que tiene dos Byte, con cuya ayuda se puede comprobar la validez de las informaciones (campo CRCID).

A continuación viene un hueco, para poder darle un pequeño descanso a la controladora hasta el comienzo de la zona de datos. Esta pausa la necesita para leer las informaciones del campo ID, y saber si tiene ante sí el sector buscado. Además, este hueco le ofrece la posibilidad de sincronizarse de nuevo.

A continuación los datos en sí, en la llamada zona de datos. Les precede de nuevo un byte de identificación (campo AM), que los identifica como tales. Después vienen los 512 Byte de datos (campo DATA). Y para que la controladora pueda verificar si los datos han variado, siguen dos Byte de corrección (campo ECC-DATA).

Y para que a la controladora le quede algo de tiempo para comprobar los Byte de corrección, de nuevo un hueco que cierra el sector, pero en el que se pueden encontrar esta vez Byte con el valor 4Eh. En total, un sector ocupa de esta forma 570 Byte en el disco duro.

Pistas extra

Pero Además de las informaciones de sectores, el disco duro acoge otro tipo de informaciones y zonas. Así por ejemplo «pistas servo», en las que se guardan cambios de flujo según un esquema determinado. Leyendo estas pistas, la electrónica se sincroniza al pulso de datos, que es necesario para la correcta comprensión de las informaciones guardadas, especialmente en el procedimiento RLL.

Algunos discos contienen además «pistas de reserva», cuya existencia no se le notifica al BIOS, y que por ello no se pueden utilizar para almacenar datos. Por ello pueden servir como reserva para la sustitución de sectores defectuosos.

Y finalmente existe, en el borde, una zona de aparcamiento, a la que se retiran los cabezales de lectura/escritura al apagar el disco duro. Esto «auto-aparcado» hoy en día pertenece al estándar, y evita los llamados Head-Crash, que antes aparecían frecuentemente durante el transporte de ordenadores, si no se habían aparcado los cabezales con ayuda de programas de utilidades especiales..

Tiempos de acceso y su medición

El rendimiento de un disco duro no se mide principalmente en su tamaño o capacidad, sino por su tiempo de acceso. Los fabricantes de hardware y los de PC indican alegremente números en este sentido, que habitualmente se mueven entre 10 y 30 milisegundos y que se denominan como velocidad de acceso del disco duro.

Pero este valor sirve más que nada para «chulear» entre los usuarios («el mío es más rápido»), sin que represente un dato realista sobre la velocidad del disco duro o la controladora. Se trata simplemente del llamado Average-Seek-Time, el tiempo medio de acceso que se necesita entre dos accesos de archivo. Esto representa el valor medio, que el fabricante ha averiguado después de múltiples mediciones según su parecer.

Aparte de esto, la mayoría de los fabricantes de discos duros indican dos valores más, que también se miden en milisegundos: el Track-to-Track-Seek-Time y el Maximum-Seek-Time.

En el caso del Track-to-Track-Seek-Time existe un poco de confusión de términos, ya que para unos denomina el tiempo de conmutación entre dos cilindros (y con ello entre cabezales), mientras que otros describen con ello el tiempo, que se necesita para desplazar el brazo de lectura/escritura a la siguiente pista. Para ser sinceros, nosotros tampoco lo sabemos con exactitud.

En cualquier caso, está mucho más claro el significado de la Maximum-Seek-Time. Describe el peor de los casos, cuando el brazo de lectura/escritura se ha de mover por todo el disco, desde la primera hasta la última pista.

Los factores enumerados tienen una influencia importante pero no decisiva sobre la velocidad definitiva del disco duro en el acceso a archivos a nivel del DOS. Ya que, ¿de qué sirve un disco duro rápido, si la controladora no es capaz de seguirlo, o si el disco está tan fragmentado que constantemente se ha de saltar de una zona a otra? Además se ha de contabilizar el tiempo, que se emplea en el camino a través de los diferentes niveles del programa de usuario, a través del DOS y sus controladores de dispositivos, el BIOS, controladores especiales de disco duro, llegando hasta la programación de la controladora del disco duro.

Quien no se fíe de las indicaciones del fabricante o que quiera comprobar el rendimiento de su disco duro bajo condiciones reales, puede echar mano de los programas de medición correspondientes. Existen, por ejemplo, en forma del programa SystemInfo (SI) de las Norton Utilities. Pero es más conocido el programa CORETEST del fabricante americano de disco Core International.

Estos programas miden habitualmente la tasa de transferencia de datos, leyendo un bloque de datos lo más grande posible del disco duro. Sin embargo, el tamaño del bloque queda limitado al tamaño de un cilindro completo, para que el brazo de lectura/escritura no se haya de mover durante el acceso. Y precisamente aquí estriba el problema, que hace poco fiable los resultados de medición, sobre todo en conexión con discos ESDI, SCSI e IDE. Ya que en el momento en que la controladora asociada le oculta al BIOS las verdaderas «medidas» del disco duro, y le simula uno de los formatos conocidos, no se puede excluir un cambio de pista durante estos accesos. Pero este trae consigo un retraso inesperado, y naturalmente distorsiona los resultados de las medidas en contra del disco duro.

Si a continuación se quiere medir explícitamente la duración de un cambio de pista, se encuentra uno con el mismo problema. Ya que este cambio de pista sólo es posible leyendo dos sectores en pistas contiguas, la conversión interna de las direcciones de segmento del disco puede llegar a provocar que en realidad no sea necesario un cambio de pista. Por ello en pantalla aparece el tiempo de cambio de pista, 0, ya que los sectores se pudieron leer en secuencia.

Controlador caché y programas caché

Especialmente poco fiables son los resultados de estos programas de test, cuando en el sistema del ordenador hay implementados programas caché, ya sea en la parte de software, como en la de hardware. En el nivel del software, esto lo realizan programas como el SMARTDRV de Microsoft o el conocido PCQuick. Se enganchan en la interrupción de disco duro del BIOS, y por ello intervienen las llamadas de lectura y escritura del DOS.

Las peticiones de lectura se pasan normalmente a la controladora del disco duro, pero a la vez guardan las informaciones leídas en un buffer de caché propio, antes de devolverlas al usuario.

Según el tamaño del buffer de caché reservado, existen una cantidad mayor o menor de sectores leídos en el buffer del caché. Si el programa de caché descubre, durante una nueva petición de lectura que el sector indicado aún se encuentra en la memoria caché, no ha de molestar al disco duro, sino que puede devolver las informaciones directamente de la memoria caché. Esto naturalmente acelera el acceso en gran medida, sin que el programa de test se diera cuenta de la falta de acceso al disco.

Pero con un par de trucos, se puede evitar este tipo de problema. El programa de test ha de analizar durante su arranque la interrupción de disquete del BIOS 13h, y comprobar si sigue apuntando a la ROM BIOS. Muy probablemente hay activo un programa de caché, y el programa de test puede pedirle al usuario que desinstale o desactive este programa, antes de comenzar con los tests.

Pero estos mecanismos de caché se convierten en totalmente invisibles, si se implementan a nivel de hardware. Así por ejemplo hace tiempo que existen controladoras de disco duro, que disponen de esta función y que tienen una memoria caché que puede llegar hasta los 16 MB. En este caso la petición de lectura no se interviene a nivel del BIOS, sino a nivel de hardware, con lo que se puede engañar completamente el programa de control y test.

Particiones de disco duro

Quien haya instalado alguna vez un disco duro por si mismo, o que haya ampliado su PC con el sistema operativo XENIX o OS/2, ha tenido que utilizarlo. Estamos hablando de FDISK, el programa de particionamiento de discos duros de MS-DOS. No sólo es la llave para utilizar discos duros de alta capacidad, sino también para la instalación paralela de varios sistemas operativos en un ordenador.

El proceso de formateado

FDISK en realidad sólo representa una parte de este proceso de formateado que tiene tres niveles, y en cuyo transcurso se particiona el disco duro, preparándolo para acoger uno o varios sistemas operativos. En el primer paso se realiza el formateo a bajo nivel, que divide el disco duro en pistas (cilindros) y sectores, escribiendo las marcas de direcciones correspondientes en el disco duro. Este (pre) formateo es necesario, por que los discos duros son «soft-sectored», como se diría en el caso de un disquete. Las marcas de direcciones, que se necesitan para la identificación de los diferentes sectores no se ha colocado en el disco duro por parte del fabricante.

Los poseedores de un PC compatible XT aún se acordarán de que el formateo a bajo nivel en estos aparatos se había de iniciar con ayuda del programa DOS DEBUG, llamando a través de DEBUG una rutina en el BIOS ampliada del disco duro. Hoy en día, el formateo a bajo nivel ya no es tan dificultoso, ya que la mayoría de los fabricantes añaden a sus unidades los programas correspondientes, que realizan este trabajo automáticamente.*

Partición del disco duro

En el siguiente paso, se realiza la partición del disco duro, que se asemeja a una división en diferentes zonas, claramente diferenciadas. Su tarea original era la de preparar varias zonas de disco duro, que podían pasar a posesión de los diferentes sistemas operativos y ser gestionados por ellos, sin que las diferentes estructuras organizativas de los diferentes sistemas llevara a conflictos.

A causa de la bajada de precios en el ámbito del hardware de PC, resultó otra razón para la partición de un disco duro, ya que ahora se convirtieron en asequibles los discos duros con capacidades por encima de los 10 y 20 MByte, los de 40, 80 o varios cientos de MByte. Pero estos no podían ser aprovechados totalmente por el DOS, al menos hasta la versión 3.3, ya que por una parte sólo gestionaba almacenamientos masivos hasta 32 MByte, y por otra parte no estaba en condiciones de dividir un disco duro en varias unidades utilizables por el DOS.

Esta falta se eliminó con la versión 3.3 del DOS, que seguía manteniendo el límite de capacidad de 32 MByte, pero que permitía una «extended partition» aparte de la «primary partition», que a su vez se podía dividir hasta en 23 dispositivos lógicos. Ya que cada uno de estos dispositivos en la partición extendida podía acoger hasta 32 MByte, la capacidad de disco duro soportada bajo DOS 3.3 se aumentó hasta 768 MByte.

Aún más allá va la versión 4.0 del DOS, que gracias a algunas modificaciones en el ámbito de los controladores de dispositivos, soporta almacenamientos masivos hasta 2 GByte. A pesar de ello, muchos usuarios siguen prefiriendo la partición del disco duro en unidades más pequeñas (pronúnciese unidades lógicas), ya que esto resulta en un mejor control, que trabajando con una unidad grande, en la que se pueden colocar muchos cientos o incluso miles de archivos.

Mientras que la partición primaria se ha de encontrar en los primeros 32 MByte del disco duro, la partición extendida puede colocarse en cualquier punto. FDISK marca este tipo de particiones con los nombres de «PRI DOS» y «EXT DOS».

El sector de particiones

Punto de partida para la partición de un disco duro bajo todas las versiones de DOS, es el sector de particiones, que FDISK crea durante su primera llamada en el primer sector del disco duro (cabezal 0, cilindro 0, sector 0). Este es el sector, y no el sector de arranque del DOS, que carga el BIOS después de un Reset o un arranque del sistema a la posición de memoria 0000:7C00, siempre que no se encuentre un disquete en la unidad A:.

Si en los dos últimos Byte de los 512 de este sector se encuentra con la secuencia de códigos 55h, AAh, considera a este sector como ejecutable, y comienza la ejecución de programa con el primer byte del sector. De lo contrario, el BIOS da un mensaje de error y se va, según versión o fabricante, a ejecutar el ROM -BASIC o a un bucle sin fin.

El código de programa que hay en el sector de arranque es el que tiene como tarea el reconocer la partición activa, y con ello el sistema operativo a ejecutar, cargar su sector de arranque y comenzar la ejecución del código de programa que allí está contenido. Ya que este código de programa, por definición, también se ha de encontrar en la posición de memoria 0000:7C00, el código de partición primeramente se desplaza a la posición de memoria 0000:0600 y con ello deja espacio para el sector de arranque.

Tabla: Estructura del sector de particiones de un disco duro

Direcc.	Contenido	Tipo
+000h	Código de partición	Código
+1BEh	1. Entrada en la tabla de particiones	16 Byte
+1CEh	2. Entrada en la tabla de particiones	16 Byte
+1DEh	3. Entrada en la tabla de particiones	16 Byte
+1EEh	4. Entrada en la tabla de particiones	16 Byte
+1FEh	Código de identificación (AA55h), que define el sector de particiones como tal	2 Byte

Longitud: 200h (512 Byte)

La tabla de particiones

Dónde se encuentra el sector de arranque a cargar en el disco duro y a qué partición pertenece, la rutina lo puede averiguar en el contenido de la tabla de particiones, que se encuentra en la dirección 1BEh en el sector de partición.

Tabla : Estructura de una entrada en la tabla de particiones

Direcc.	Contenido	Tipo
+00h	Estado de partición 00h = inactiva 80h = Partición de arranque	1 BYTE
+01h	Cabezal de lectura/escritura con el que comienza la partición	1 BYTE
+02h	Sector y cilindro, con el que comienza la partición	1 WORD
+04h	Tipo de partición 00h = Entrada no ocupada 01h = DOS con 12-bit-FAT (primary Part.) 02h = XENIX 03h = XENIX 04h = DOS con 16-bit-FAT (primary Part.) 05h = extended DOS-Partition (a partir de DOS 3.3) 06h = Partición DOS-4.0, con más de 32 MB DBh = Concurrent DOS Son posibles otros códigos en unión con otros sistemas operativos o algunos controladores de software.	1 BYTE
+05h	Cabezal de lectura/escritura con el que termina la partición	1 BYTE
+06h	Sector y cilindro, con el que termina la partición	1 WORD
+08h	Distancia del primer sector de la partición (Sector de arranque) del sector de part. en sectores	1 DWORD
+0Ch	Número de sectores en esta partición	1 DWORD

Longitud: 10h (16 Byte)

Cada partición se describe en la tabla mediante una estructura, que se compone de 16 Byte. ya que la tabla casi se encuentra al final del sector de partición, solo le queda espacio para 4 entradas, lo que naturalmente limita el número de particiones a 4. Para poder alojar más de 4 particiones en un disco duro, algunos fabricantes de discos duros disponen de programas de configuración especiales, que desplazan hacia adelante la tabla en el sector de particiones, y que simultáneamente instalan un nuevo código de particiones, que accede a la tabla desplazada.

Normalmente también se dispone en estos casos de una «Multiple-Boot-Option», con cuya ayuda el usuario puede decidir durante la ejecución del código de programa, qué partición debe ser arrancada. Con ello se pueden hacer servir varios sistemas operativos, seleccionando en cada arranque del ordenador, aquel que necesitamos.

En la estructura fundamental sin embargo, la mayoría de estos programas no modifican nada. Tenga sin embargo en cuenta, que las diferentes entradas de partición no siempre comienzan con la primera entrada de la tabla. Por ejemplo es posible, que una partición de un disco duro no se describa por la primera entrada de la tabla, sino por la segunda, tercera o incluso la cuarta.

Arranque de la partición de arranque

Cual de las diferentes particiones es la partición de arranque, se puede reconocer en el primer campo de la estructura de partición correspondiente. El valor 00h está por «no activa», mientras que el valor 80h indica la partición a arrancar. Si el código de partición durante el análisis descubre que no hay, o que hay más de una partición para arrancar, o incluso un código desconocido, el proceso de arranque se interrumpe con un mensaje de error y termina en un bucle sin fin, que sólo se puede abandonar mediante un Reset.

Si el código de partición reconoce la partición a arrancar, puede averiguar la posición de esta partición en el disco duro, con ayuda de los dos campos siguientes. El número de sector y cilindro se encuentran codificados de la misma forma que los espera la interrupción 13h del BIOS (disquete/disco duro). Los bits 6 y 7 del número de sector representan los bits 8 y 9 del número de cilindro. Esta

coincidencia no es aleatoria, ya que en este momento la interrupción 13h y sus funciones representan la única posibilidad de acceder al disco duro. (Las funciones DOS aún no están disponibles, ya que el DOS aún se ha de arrancar).

Otras informaciones en la estructura de arranque

A pesar de que estas informaciones para cargar el sector de arranque de la partición a arrancar son más que suficientes, la tabla de particiones contiene otras informaciones, que después son importantes para modificaciones y ampliaciones posteriores. A continuación del sector de arranque se encuentra un campo, que describe el tipo de sistema operativo, que se esconde detrás de la partición. La figura anterior le muestra los diferentes códigos, que se pueden encontrar en este campo.

Aparte del sector de arranque, en el sector de particiones también se indica el último sector de la partición. La posición del sector se describe de nuevo mediante la indicación de cabezal, cilindro y número de sector. Además puede encontrar en los últimos dos campos de una entrada de la tabla la cantidad de los sectores en la partición, y la distancia del sector de arranque de la partición del sector de particiones, contada en sectores.

Si se analiza la tabla de particiones, habitualmente se puede determinar que la primera partición no comienza inmediatamente detrás del sector de particiones, sino en el primer sector de la pista cero en el segundo cabezal del lectura/escritura. Así que casi se desperdicia toda la pista cero del primer cabezal, si no se tiene en cuenta el sector de partición del primer sector de esta pista. Y aparte de esto, hay algunas otras incoherencias, en cuanto a lo que se refiere las particiones extendidas del DOS.

Estructura de la partición extendida bajo DOS

Para empezar, DOS 3.3 sólo permite una partición extendida en el disco duro, aparte de la partición primaria. Pero FDISK le da de nuevo un sector de particiones, que no contiene código de programa, pero sí, como siempre, una tabla de particiones. Ésta se compone habitualmente de dos entradas, de las cuales la primera describe al primer dispositivo lógico de la partición extendida, y que tiene como valor de partición el valor 1 o 4 (partición DOS con FAT de 12 o 16 bits). La segunda entrada sin embargo apunta al siguiente dispositivo lógico en la partición extendida, siempre y cuando este exista.

Para soportar otros dispositivos lógicos, esta estructura se repite con cada dispositivo adicional. Con ello se crea una especie de lista enlazada, que sólo se termina cuando dentro del sector de particiones de una partición extendida el campo «tipo de partición» contiene un cero en la segunda entrada de la tabla.

Análisis de la estructura de particiones

Para que se pueda hacer una idea de la estructuración de su propio disco duro, a continuación hemos incluido dos programas, en Pascal y C, que le comentan el contenido del sector de particiones, y que siguen los sectores de particiones de las particiones extendidas a través del disco duro (si existen). Por defecto, los programas acceden al primer disco duro con el número 0, pero indicando otro número (1,2,3, etc.) al llamar a los programas, se pueden indicar otros discos duros.