(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 522 750 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
19.05.1999 Bulletin 1999/20

(51) Int Cl.6: **G11B 20/12**, G11B 20/18

(21) Application number: **92305868.9**

(22) Date of filing: **25.06.1992**

(54) **A data recording disk apparatus**

Vorrichtung zur Datenaufzeichnung auf einen Disk

Appareil d'enregistrement de données sur un disque

(84) Designated Contracting States:
DE FR GB

(30) Priority: **10.07.1991 US 727680**

(43) Date of publication of application:
13.01.1993 Bulletin 1993/02

(73) Proprietor: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventors:
- **Best, John Stewart**
  San Jose, California 95120 (US)
- **Hetzler, Steven Robert**
  Sunnyvale, California 94086 (US)

(74) Representative: **Burt, Roger James, Dr.**
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester Hampshire SO21 2JN (GB)

(56) References cited:
EP-A- 0 420 211
US-A- 4 513 392
US-A- 4 839 878

EP 0 522 750 B1

Description

[0001] This invention relates to a data recording disk apparatus (hereinafter also called "disk file").

Background of the Invention

[0002] All disk files require some means of determining the radial position of the READ-WRITE heads over the disks so that the heads can be accurately positioned over any desired track. Typically this is accomplished by placing servo information on one or more of the disk surfaces for reading by magnetic or optical heads. Some disk files, known as dedicated servo files, contain servo information only on a dedicated surface of one disk in the disk stack. In contrast, some modern files, known as sector servo files, store the servo information interspersed with the data on each disk surface. This latter approach represents the direction in which the technology is progressing and is preferred because it can be implemented at low cost and without additional componentry beyond that required to store the data and further because it provides the servo information at the data surface being accessed, thereby eliminating many mechanical and thermal sources of track misregistration. Fixed block architecture (FBA) is a common configuration used to format both dedicated servo disk files and sector servo disk files. In an FBA formatted disk file, each disk track is divided into a number of equal-sized sectors, and each sector is divided into regions containing servo information identification information (ID), and data.

[0003] A typical FBA sector servo sector format according to the prior art, for example EP-A-0 439 278 (not prepublished), is illustrated in Fig. 1A. The sector is divided into three regions: servo and recovery region 10, ID region 11 and data region12. Generally, servo and recovery region 10 contains overhead associated with sectoring, as well as servo information for sector servo files. Also, this region marks the beginning of the sector. Specifically, servo region 10 contains WRITE-to-READ recovery (W-R) and speed compensation field 13, which is used to switch the data channel from WRITE to READ and accommodate spindle speed variations; address mark (AM) field 14; and servo position field 15 containing a position-error-signal (PES). Data region 12 contains the user data, as well as overhead fields. READ-to-WRITE recovery (R-W) and speed field 19 performs a function like that of W-R and speed field 13. Voltage controlled oscillator sync (VCO) field 20 is used to synchronize the read clock with the read data. Data field 21 contains the user data and associated error checking and correction (ECC) information. ID region 11 is used by the disk data controller (DDC) to identify the physical sector number. It contains R-W and speed field 16, VCO synch field 17 and identification/error handling (ID/EH) field 18, which contains the identification information including the logical sector number.

[0004] A typical FBA dedicated servo sector format according to the prior art is illustrated in Fig. 1B. The data disk sector shown is divided into two regions: ID and recovery region 11' and data region 12'. ID and recovery region 11' contains substantially the same information as servo and recovery region 10 and ID region 11 of Fig. 1A, with the exception that servo position field 15 and R-W and speed field 16 are removed. In a dedicated servo scheme, servo position field 15 is not needed because the position information is contained on a separate surface; R-W and speed field 16 is not needed because address mark field 14' can be written whenever the succeeding VCO synch field 17' and ID/EH field 18' are written. Data region 12' contains substantially the same information as data region 12 of Fig. 1A.

[0005] ID regions 11 and ID and recovery region 11' of the prior art FBA sector formats perform two important functions. First, they uniquely identify the logical sector number of the sector in which they are located. Second, they indicate whether the sector is good or bad (that is, whether the sector can be successfully written to and read from). Unfortunately, in both the dedicated servo and sector servo scheme, the ID region requires a great deal of disk space to perform its functions. Since the disk uses the information in the ID region to perform a logical-to-physical sector conversion in order to locate the physical position on the disk corresponding to the requested logical sector, the ID region must be large enough to avoid misidentifying sectors. Furthermore, since a phase synchronous clock is typically used to read the ID/EH field, the VCO field is required; this field is typically much larger than the ID/EH field. Finally, in the sector servo scheme, due to the region transition an R-W and speed field is required to switch the data channel from READ to WRITE and accommodate spindle speed variations.

[0006] In sum, due to overhead and accuracy requirements, the ID region of the prior art FBA sector accounts for a substantial portion of the overall sector length (typically about 4%), a portion that could otherwise be used for data storage. However, the importance of its two functions has hampered efforts to eliminate it from the FBA sector.

[0007] One approach that has been used to eliminate the ID region is to encode the ID information into the data field. This eliminates the overhead associated with synching (fields 16 and 17), and allows the ECC to provide error checking for the ID information. However, it does not eliminate or reduce the actual ID information as contained in ID/EH field 18. Also, it adds a minimum of one sector time to the latency during WRITE, since the disk file controller cannot determine the good/bad sector relationship until it has read at least one sector. Further, this scheme is not well suited to compact and low-cost disk files, where the error correction time is typically quite long, since it means a revolution may be lost in the case of a sector where any bit is bad in the entire field containing the combined ID and data. Thus, while the approach of combining the ID and data regions reduces the disk space required for ID information, this approach introduces performance penalties that restrict its usefulness.

[0008] In both US-A-4513392 and EP-A-0420211, are illustrated sector formats for disk data storage apparatus requiring only two fields, the first containing identification and other control information and the second containing user data. The identification information includes physical identification of the sector by track number and sector number within the track. In US-A-4 513 392 a start-of-track indication is present in the form of a notch in the edge of the platter. There is however no discussion of how user data may be identified and correctly accessed if a sector should be defective so that its data field is incapable of storing user data.

[0009] EP-A-0 341 852 shows similar physical identification of sector addresses in a sector servo format also including servo positioning information in a header region. The problems of defects is addressed by recording a bad sector bit map in every data region. However, this requires further storage space on the disk surface. This patent does not discuss the identification of blocks of user data by logical sector number or how such logical sector numbers can be converted to the physical sector numbers where such user data is actually located so that the required data my be correctly and directly accessed.

[0010] It is therefore an object of the present invention to overcome the above drawbacks of the prior art.

[0011] According to the present invention there is provided a disk data storage apparatus including a disk, the apparatus being arranged to store blocks of data sequentially in at least one track on a surface of such disk in substantially fixed length sectors each identifiable by a respective physical sector number, each sector comprising a data region and a location region containing information for locating data, said disk surface potentially including one or more defective data regions unavailable for data storage; the apparatus further including head means for reading data from or writing data to said disk; a controller responsive to the supply of a unique logical sector number identifying one of said blocks of data to locate said head means at the physical sector in which said identified block is stored; the apparatus being characterised by : a defect map containing information related to the location of any such defective data regions on said disk; and said controller being arranged to determine a physical sector number corresponding to said supplied logical sector number from said defect map and to locate said head means at said corresponding physical sector using physical sector identifying information on said disk, said physical sector identifying information including a start-of-track indication on said at least one track, track number information and a sector mark in each location region, said controller including means for detecting said start-of-track indication and means for counting said sector marks to uniquely identify a physical sector.

[0012] The foregoing and other object, features and advantages of the invention will be apparent from the following more particular description of the preferred embodiment of the invention, as illustrated in the accompanying drawings.

Brief Description of the Drawings

[0013] Fig. 1A is a schematic diagram illustrating a fixed block architecture sector servo sector format according to the prior art.

[0014] Fig. 1B is a schematic diagram illustrating a fixed block architecture dedicated servo sector format according to the prior art.

[0015] Fig. 2A is a schematic diagram illustrating a fixed block architecture sector servo sector format according to the present invention.

[0016] Fig. 2B is a schematic diagram illustrating a fixed block architecture dedicated servo sector format according to the present invention.

[0017] Fig. 3 is a table illustrating a bad sector list, a logical-to-physical mapping, and a look-up table according to the present invention.

[0018] Fig. 4 is a C language procedure illustrating an algorithm for creating a look-up table according to the present invention.

[0019] Fig. 5 is a flowchart illustrating an algorithm for performing logical-to-physical sector conversion according to the present invention.

Detailed Description of the Invention

IA. NO-ID FBA SECTOR SERVO SECTOR FORMAT

[0020] Fig. 2A illustrates in schematic diagram form an FBA sector servo sector according to the present invention. Servo and recovery region 30 includes W-R and speed field 33, AM field 34 and position field 35. Position field 35 itself includes track number subfield 36, EH subfield 37 and PES subfield 38. Data region 32 includes R-W and speed field 39, VCO synch field 40 and data field 41.

[0021] In operation, fields 33, 34, 39, 40 and 41 provide substantially the same functions as fields 13, 14, 19, 20 and 21 shown in prior art Fig. 1A and described previously. Fields 16, 17 and 18 from prior art Fig. 1A are eliminated.

[0022] Position field 35 is expanded over position field 15 of the prior art, as shown in the lower portion of Fig. 2A. According to the preferred embodiment of the present invention, the full track number corresponding to the track on which the sector is located is recorded in track number subfield 36. EH subfield 37 is provided as an adjunct to the full track number to ensure that the track number has been read correctly. Finally, the actual position-error-signal is encoded in PES subfield 38. This subfield contains substantially the same information as position field 15 of prior art Fig. 1A.

[0023] It is to be noted that both the full track number and the EH information may be recorded according to any of a variety of techniques well known to those skilled in the art of data recording disk sector formats. For instance, a gray code may be used to record the full track number because it allows the full track number to be accurately read when the disk head is off-track. The EH information may implement any of a variety of schemes well known to those skilled in the art, including error checking and correction (ECC) codes and cyclic redundancy check (CRC) codes. According to the preferred embodiment of the present invention, EH subfield 37 is implemented using CRC, an error-check-only code. CRC is chosen because it is easy to compute and because it requires a small number of bits (roughly 16). Since CRC bits will not form a gray code, EH subfield 27 is implemented in a code that is accurately readable only when the head is on track. This implementation has no adverse impact on the present invention, since an absolutely accurate reading of the track number subfield is required only when the head is already on-track.

[0024] The no-id sector format of the present invention provides several space saving advantages over the prior art. The track number need not be redundantly encoded in an ID region but instead is communicated to the DDC by the servo system at runtime. Track number subfield 36 and EH subfield 37 are substantially shorter than ID region 11 shown in prior art Fig. 1A, since the FBA format of the present invention is able to use CRC rather than other forms of ECC and since the maximum track number is one or two orders of magnitude smaller than the maximum sector number encoded in prior art ID/EH field 18. Since fewer bits are required to represent the track number, this quantity may be recorded at lower density, so that track number subfield 36 and EH subfield 37 can be read without a phase synchronous clock; therefore VCO synch field 17 is not required. Finally, since no separate ID region is set up, R-W and speed field 16 is eliminated.

[0025] As discussed above, the preferred embodiment of the present invention contemplates the use of a full track number in track number subfield 36 in order to provide absolute identification of the track number for READ and WRITE operations. However, it is to be understood that the track number may be encoded according to a variety of alternative schemes. For instance, PES subfield 38 may itself assume the form of a pattern which is unique over a range of tracks, and may be used to construct the LSBs of the track number. Thus, if the PES pattern repeats every four tracks, it may be used to provide the two LSBs of the track number. Or the MSBs of the track number may be omitted altogether, and the servo electronics depended on to determine which half (or quarter, or eighth) of the disk the head is positioned over.

[0026] The application of the present invention to sector servo disk files has been described with reference to the sector servo sector format shown in Fig. 2A. However, it should be noted that the present invention may be applied to any sector servo format in which the track number may be encoded in the servo and recovery region. For example, instead of a fixed spacial relationship between the servo regions and data regions, a sector servo format may provide servo information at other than fixed intervals with respect to data, such as within the data region itself. Such a scheme is disclosed in US Patent application serial number 07/466,195, filed January 1, 1990. In this case, the fields in Fig. 2A may be repositioned such that servo and recovery region 30 is split into separate recovery and servo regions with the recovery region containing fields 33 and 34 and the servo region containing field 35. The sector number and track number may then be determined in the same manner as with the sector format shown in Fig. 2A.

IB. NO-ID FBA DEDICATED SERVO SECTOR FORMAT

[0027] Fig. 2B illustrates in schematic diagram form an FBA data disk dedicated servo sector according to the present invention. Track ID and recovery region 30' is produced by combining aspects of prior art ID and recovery region 11' with the full track number. However, it contains no position error information analogous to that found in PES subfield 38. Track ID and recovery region 30' includes W-R and speed field 33', AM field 34' and track field 35'. Track field 35' itself includes track number subfield 36' and EH subfield 37'. Data region 32' assumes the same form as data region 32 in the sector servo format.

[0028] In operation, fields 33', 34', 39', 40', and 41' provide substantially the same functions as fields 13', 14', 19', 20', and 21' shown in prior art Fig. 1B and described previously. Fields 17' and 18' from prior art Fig 1B are eliminated. Track number subfield 36' and EH subfield 37' are identical to fields 36 and 37 in Fig. 2A. The full track number is therefore provided in the same manner as for the sector servo format, including the alternative encodement schemes discussed in section IA. As with the sector servo format, the track number field must be accurately readable only when the head is on-track, since it is used only during read and write operations.

[0029] While Fig. 2B and the above description have focused on a dedicated servo data disk having the sector format of the present invention, it is to be noted that this sector format may also be applied to the dedicated servo disk itself. That is, track field 35' may be encoded on the dedicated servo disk. This configuration is not the preferred embodiment, as it requires more rigorous constraints on the system mechanics. However, it has the advantage of lower overhead since field 35' is not placed on the data srface, and thus may provide an attractive alternative that is within the scope of the present invention.

[0030] Dedicated servo systems record the head positioning information on a single magnetic recording surface, separate from the data surface(s). However, it is to be understood that there are many other types of servo systems, which for the purposes of the present invention function similarly to the dedicated servo system. These include systems such as: optical recording continuous groove systems, where the tracking information is provided by grooves in the substrate; buried servos, whether magnetically, capacitively, or otherwise encoded; optical tracking servos, where the servo information is optically encoded on a data or servo surface; and baseplate reference servo systems, where the positioning information is provided by some external reference such as an optical encoder or a stepping motor. Finally, it is worthy of note that the present invention may be readily applied to systems combining aspects of sector servo and dedicated servo.

II. LOGICAL-TO-PHYSICAL SECTOR CONVERSION

[0031] According to the prior art, physical sectors are accessed by counting actual sectors. Physical sector 0 refers to sector 0, head 0, track 0. The number increases by counting sectors around a track, then switching heads, then tracks.

[0032] According to the present invention, physical sectors are accessed by counting start-of-sector indicating marks after a start-of-track indicating mark. A start of track indicating mark is encoded in a preselected track to provide a reference point from which to count sectors. A start-of-sector indicating mark is encoded in each sector. In operation, a requested physical sector is accessed by detecting the start-of-track indicating mark and then counting start-of-sector indicating marks until the appropriate sector is reached.

[0033] In the preferred embodiment of the present invention, the start of track indicating mark takes the form of a unique bit called an index mark, recorded in a preselected track number subfield 36. AM field 34 of each sector is used as the start-of-sector indicating mark. The servo system then counts AM fields after an index mark to locate a requested sector. No additional overhead is incurred, since the servo system must locate every AM field in order to read the PES subfield. Also, the scheme requires only a small counter, since the number of sectors per track is typically less than 64, and since the sector count is reset each time an index mark is detected.

[0034] There are a number of variations on the embodiment that serve to improve error tolerance where AM field 34 is missed due to a disk defect. First, the physical sector number may be encoded into track number field 36. Second, an open loop timer may be used to provide a timeout error if an address mark is not detected within a given time interval. Third, position fields may be counted in addition to AM fields.

III. LOOK-UP TABLE

[0035] Prior art disk file implementations handle bad sectors via the sector ID region. According to the present invention in which the ID region has been eliminated, bad sectors are identified and mapped out of the data recording disk file during logical-to-physical sector conversion using a look-up table. The table is created at format time, and consists of entries describing clusters of bad sectors (where a cluster is defined as a sequential group of physical bad sectors) by starting location and quantity. In operation, a requested logical sector location identifier is referenced into the table, and a corresponding entry is found describing an offset to the physical sector location identifier.

[0036] More specifically, at format time, the data recording disk file generates a list of physical sector location identifiers for the bad sectors in the disk file. This list may be generated using any of the various techniques known in the art for locating bad sectors. The list is held in RAM and is used to construct the look-up table. Fig. 3 illustrates in tabular form an example of a bad sector list, a logical-to-physical mapping and a look-up table according to the preferred embodiment of the present invention. The bad sector list indicates that physical sectors 3, 5, 6, 9, 10, 11 and 13 were found to be bad at format time. The logical-to-physical mapping shows the desired mapping to be achieved by the look-up table. Logical sector 1 will be found at physical sector 1; logical sector 2 will map to physical sector 2; logical sector 3 will map to physical sector 4 since physical sector 3 is defective, etc.

[0037] The look-up table shows the result of compressing the bad sector list in accordance with the present invention. Starting at logical sector 3, there is a cluster of 1 bad sector; thus, sectors 3 and beyond must be offset by 1 to map into the correct physical sector. Starting at logical sector 4, there is a cluster of 2 bad sectors; thus, sectors 4 and beyond must be offset by a total of 3 to map into the correct physical sector. Starting at logical sector 6, there is a cluster of 3 bad sectors; thus, sectors 6 and beyond must be offset by a total of 6 to map into the correct physical sector. Finally, starting at logical sector 7, there is a cluster of 1 bad sector; thus, sectors 7 and beyond must be offset by a total of 7 to map into the correct physical sector.

[0038] It is to be noted that the look-up table may be stored either on the boot track of the disk or in some non-volatile semiconductor memory in the data recording disk file, such as EEPROM, FRAM, or PROM. The first column of the table must be wide enough to hold the largest physical sector number (24 bits is sufficient in most cases). The second column must be wide enough and the table long enough to hold the maximum number of bad sector clusters likely to be encountered (16 bits is sufficient in most cases).

[0039] Fig. 4 illustrates in algorithmic form a procedure for creating a look-up table according to the preferred embodiment of the present invention. The look-up table generation algorithm (shown in C programming language) processes the bad sector list as follows. First, the list of bad sectors is converted into the compact table representation. Then the first "while" loop computes the offset for the first block by determining its size (i.e. the number of contiguous bad sectors). Next, the second "while" loop performs the same operation for all the remaining blocks, determining each offset by adding the number of sectors in the block to the total number of bad sectors prior to the block. Finally, the look-up table is adjusted to contain an odd number of entries, for ease of access by a binary search look-up routine. Of course, many other search types are possible, the use of which would obviate this final step. Also, it is worthy of note that the look-up table generation algorithm may be executed against the entire bad sector list, or, where storage or processing constraints arise, may be executed in a piecewise fashion, so that each block of bad sectors is processed as it is located.

[0040] Fig. 5 illustrates in flowchart form an algorithm for performing logical-to-physical sector conversion according to the preferred embodiment of the present invention. A READ or WRITE request, including a logical sector location identifier, is generated within the disk file. A look-up table containing bad sector information and indices as described above is accessed and searched for the largest logical sector entry less than or equal to the requested logical sector. In the event no entry is found, the physical sector location identifier is equal to the logical sector location identifier, and the requested logical sector is accessed by computing head, track and sector number directly from the logical sector location identifier according to any of a variety of techniques well known to those skilled in the art.

[0041] If an entry is found in the look-up table, the corresponding offset is extracted from the look-up table and added to the logical sector location identifier of the requested logical sector. This sum, representing the physical sector location identifier, is used to compute the head, track and sector number corresponding to the requested logical sector.

[0042] In some implementations (e.g. SCSI), it may be desirable to allow for the relocation of bad sectors without resorting to a complete device format operation. This feature is implemented in conjunction with the present invention by allocating a set of sectors as spares prior to an initial format operation. The spares are treated as though they are bad sectors for purposes of constructing the logical-to-physical mapping table. The spare sectors are then distinguished from bad sectors, for instance by means of a flag in the mapping table or a separate list. Relocation of a bad sector is accomplished by logically substituting one of the spare sectors in its place, shifting all the intervening sectors by one logical sector location, updating the spare sector list to reflect the use of the spare sector, and re-constructing the logical-to-physical mapping table to reflect all the above changes.

[0043] A look-up table constructed according to the preferred embodiment of the present invention as described above has the advantageous effect of placing most of the computational overhead into building the table during the format operation, leaving a simple computation to support logical-to-physical mappings during seek operations. It is to be noted that the preferred embodiment look-up table approach is maximally effective when the total number of bad sectors in the data recording disk file is small, since this keeps the overall table size small and thus keeps the worst-case binary search of the table short. For example, a disk file having 240K sectors and .1% bad sectors would require a table having less then 1000 entries. The look-up table, in cooperation with the above-described algorithm, provides a scheme that efficiently catalogues bad sectors and enables rapid calculation of the correct physical sector corresponding to a requested logical sector.

[0044] Furthermore, the No-ID scheme of the present invention is particularly well suited for use with disk files having different heads for reading and writing, such as magneto-resistive read heads, where the heads must be moved to slightly different radial positions for reading and writing (a practice known as micro-jog). With prior art fixed block architectures, writing a data field requires prior reading of the ID field. However, it is not possible to complete a micro-jog between the two fields, so a revolution is lost in performing the micro-jog. The No-ID format of the present invention eliminates this problem since it has no ID field to be read.

Claims

1. A disk data storage apparatus including a disk, the apparatus being arranged to store blocks of data sequentially in at least one track on a surface of such disk in substantially fixed length sectors each identifiable by a respective physical sector number, each sector comprising a data region (32) and a location region (30) containing information for locating data, said disk surface potentially including one or more defective data regions unavailable for data storage;

   the apparatus further including
   head means for reading data from or writing data to said disk; a controller responsive to the supply of a unique logical sector number identifying one of said blocks of data to locate said head means at the physical sector in which said identified block is stored;

   the apparatus being characterised by :

   a defect map containing information related to the location of any such defective data regions on said disk;

and said controller being arranged to determine a physical sector number corresponding to said supplied logical sector number from said defect map and to locate said head means at said corresponding physical sector using physical sector identifying information on said disk, said physical sector identifying information including a start-of-track indication on said at least one track, track number information (36) and a sector mark (34) in each location region (30), said controller including means for detecting said start-of-track indication and means for counting said sector marks to uniquely identify a physical sector.

2. Data storage apparatus as claimed in claim 1 in which said defect map is a lookup table containing offset information from which the offset, if any, from each logical sector number of its corresponding physical sector can be determined, said offset information corresponding to the cumulative number of defective data regions between said logical sector number and said physical sector number.

3. Data storage apparatus as claimed in claim 2 in which said lookup table contains entries only for logical sector numbers at which said offset information changes.

4. Data storage apparatus as claimed in any preceding claim in which said disk surface includes spare data regions identifiable by respective physical sector numbers which spare data regions are available to store information intended to be stored in said defective data regions, said spare sectors being treated as defective sectors in computing said defect map offsets.

5. Data storage apparatus as claimed in any preceding claim in which said disk controller stores said defect map in a non-volatile semiconductor memory.

6. Data storage apparatus as claimed in any of claims 1 to 4 in which said defect map is stored on said disk.

7. Data storage apparatus as claimed in any preceding claim in which said apparatus is arranged to store data in a plurality of tracks and said location regions are servo regions including prewritten head positioning information (35) defining the positions of said tracks, and in which said data regions are unbroken regions immediately adjacent to a respective servo region.

8. Data storage apparatus as claimed in any of claims 1 to 6 in which said apparatus is arranged to store data in a plurality of tracks and said location regions are servo regions including prewritten head positioning information defining the positions of said tracks, and in which at least one data region is split by a servo region.

9. Data storage apparatus as claimed in any preceding claim in which said apparatus is arranged to store data in a plurality of tracks, each track including a start-of-track indication.

10. Data storage apparatus as claimed in any preceding claim in which said head means includes a head for writing information on said disk and a magnetoresistive head for reading information from said disk.

Patentansprüche

1. Ein Plattenlaufwerk mit einer Platte, das so aufgebaut ist, daß Datenblöcke auf mindestens einer Spur auf einer Oberfläche der Platte sequentiell in Sektoren mit im wesentlichen fester Länge, die durch ihre physische Sektornummer identifizierbar sind, gespeichert werden, wobei jeder Sektor einen Datenbereich 32 und einen Positionsbereich 30, der die Informationen zum Auffinden der Datenposition enthält, besitzt, und wobei die Plattenoberfläche möglicherweise einen oder mehrere defekte Datenbereiche enthält, die nicht für die Speicherung von Daten zur Verfügung stehen;

wobei das Laufwerk außerdem folgendes enthält:

einen Kopf zum Lesen der Daten von der Platte und zum Schreiben von Daten auf die Platte; einen Controller, der bei der Angabe einer eindeutigen logischen Sektornummer einen der Datenblöcke identifiziert, um den Kopf auf dem physischen Sektor zu positionieren, in dem der identifizierte Block gespeichert ist;

und gekennzeichnet ist durch eine Tabelle der defekten Sektoren, die Informationen über die Position solcher defekter Datenbereiche auf der Platte enthält;

wobei der Controller die der angegebenen logischen Sektornummer entsprechende physische Sektornummer anhand der Tabelle ermittelt und den Kopf anhand der Identifikationsdaten für den physischen Sektor, die aus einer Spuranfangsmarkierung, einer Spurnummer 36 und einer Sektormarkierung 34 in jedem Positionsbereich besteht, auf den betreffenden physischen Sektor positioniert;

und wobei der Controller Mittel zur Erkennung der Spuranfangsmarkierung und Mittel zum Zählen der Sektormarkierungen besitzt, um einen physischen Sektor eindeutig zu identifizieren.

2. Ein Plattenlaufwerk gemäß Anspruch 1, bei dem die Liste der defekten Sektoren eine Referenztabelle ist, die Offsetinformationen enthält, von der ein eventuell vorhandener Offset jeder logischen Sektornummer gegenüber der zugehörigen physischen Sektornummer ermittelt werden kann, wobei der Offset der kumulativen Anzahl defekter Datenbereiche zwischen der logischen Sektornummer und der physischen Sektornummer entspricht.

3. Ein Plattenlaufwerk gemäß Anspruch 2, bei dem die Referenztabelle nur für diejenigen logischen Sektornummern Einträge enthält, bei denen sich der Offset ändert.

4. Ein Plattenlaufwerk gemäß Anspruch 1 bis 3, bei dem die Plattenoberfläche Ersatzdatenbereiche enthält, die anhand ihrer physischen Sektornummer identifiziert werden können, wobei die Ersatzdatenbereiche für die Speicherung der Informationen, die in den defekten Datenbereichen gespeichert werden sollten, zur Verfügung stehen und bei der Berechnung der Offsets in der Liste der defekten Sektoren wie defekte Sektoren behandelt werden.

5. Ein Plattenlaufwerk gemäß Anspruch 1 bis 4, bei dem der Controller die Liste der defekten Sektoren in einem nichtflüchtigen Halbleiterspeicher speichert.

6. Ein Plattenlaufwerk gemäß Anspruch 1 bis 4, bei dem die Liste der defekten Sektoren auf der Platte gespeichert wird.

7. Ein Plattenlaufwerk gemäß Anspruch 1 bis 6, das so aufgebaut ist, daß Daten in mehreren Spuren gespeichert werden und die Positionsbereiche Servobereiche sind, die vorgegebenen Kopfpositionierungsinformationen 35 enthalten, in denen Position der Spuren definiert werden, und die Datenbereiche zusammenhängende Bereiche unmittelbar im Anschluß an den betreffenden Servobereich sind.

8. Ein Plattenlaufwerk gemäß Anspruch 1 bis 6, das so aufgebaut ist, daß Daten in mehreren Spuren gespeichert werden und die Positionsbereiche Servobereiche sind, die vorgegebene Kopfpositionierungsinformationen enthalten, in denen die Position der Spuren definiert werden, und mindestens ein Datenbereich durch einen Servobereich geteilt wird.

9. Ein Plattenlaufwerk gemäß Anspruch 1 bis 8, das so aufgebaut ist, daß Daten in mehreren Spuren gespeichert werden, wobei jede Spur eine Spuranfangsmarkierung besitzt.

10. Ein Plattenlaufwerk gemäß Anspruch 1 bis 9, bei dem das Kopfmittel einen Kopf zum Schreiben von Daten auf die Platte und einen magnetoresistiven Kopf zum Lesen der Daten von der Platte enthält.

Revendications

1. Dispositif de stockage de données sur disque incluant un disque, le dispositif étant conçu pour mémoriser des blocs de données séquentiellement dans au moins une piste sur une surface d'un tel disque dans des secteurs de longueurs pratiquement fixes, chacun identifiable par un numéro de secteur physique respectif, chaque secteur comprenant une région de données (32) et une région de positionnement (30) contenant des informations pour positionner les données, ladite surface du disque incluant potentiellement une ou plusieurs régions de données défectueuses indisponibles pour la mémorisation de données ;

le dispositif incluant de plus
un moyen de tête pour lire les données à partir dudit disque ou écrire les données sur ledit disque ; un contrôleur sensible à la délivrance d'un numéro de secteur logique particulier identifiant un desdits blocs de données pour positionner ledit moyen de tête au niveau du secteur physique dans lequel ledit bloc identifié est mémorisé ; le dispositif étant caractérisé par : une carte de défaut contenant des informations liées à l'emplacement de toutes régions de données défectueuses quelconques sur ledit disque ;
et ledit contrôleur étant conçu pour déterminer un numéro de secteur physique correspondant audit numéro de secteur logique délivré à partir de ladite carte de défaut et pour positionner ledit moyen de tête au niveau dudit secteur physique correspondant en utilisant les informations d'identification de secteur physique sur ledit disque, lesdites informations d'identification de secteur physique incluant une indication de début de piste sur ladite au moins une piste, des informations de numéro de piste (36) et un repère de secteur (34) dans chaque région de positionnement (30), ledit contrôleur incluant un moyen pour détecter ladite indication de début de piste et un moyen pour compter lesdits

repères de secteur pour identifier particulièrement un secteur physique.

2. Dispositif de stockage de données selon la revendication 1, dans lequel ladite carte de défaut est une table de consultation contenant des informations décalées à partir desquelles le décalage, s'il y en a, à partir de chaque numéro de secteur logique de son secteur physique correspondant peut être déterminé, lesdites informations de décalage correspondant au nombre cumulé de régions de données défectueuses entre ledit numéro de secteur logique et ledit numéro de secteur physique.

3. Dispositif de stockage de données selon la revendication 2, dans lequel la table de consultation contient des entrées seulement pour les numéros de secteur logique auxquels lesdites informations de décalage changent.

4. Dispositif de stockage de données selon l'une quelconque des revendications précédentes, dans lequel ladite surface du disque inclut des régions de données de réserve identifiables par des numéros de secteur physique respectifs lesquelles, régions de données de réserve, sont disponibles pour mémoriser les informations destinées à être mémorisées dans lesdites régions de données défectueuses, lesdits secteurs de réserve étant traités comme secteurs défectueux lors du calcul desdits décalages de carte de défaut.

5. Dispositif de stockage de données selon l'une quelconque des revendications précédentes, dans lequel ledit contrôleur de disque mémorise ladite carte de défaut dans une mémoire à semi-conducteurs rémanente.

6. Dispositif de stockage de données selon l'une quelconque des revendications 1 à 4, dans lequel ladite carte de défaut est mémorisée sur ledit disque.

7. Dispositif de stockage de données selon l'une quelconque des revendications précédentes, dans lequel ledit dispositif est conçu pour mémoriser des données dans une pluralité de pistes et lesdites régions de positionnement sont des régions d'asservissement incluant des informations de positionnement de tête préécrites (35) définissant les positions desdites pistes et dans lesquelles lesdites régions de données sont des régions non séparées immédiatement adjacentes à une région d'asservissement respective.

8. Dispositif de stockage de données selon l'une quelconque des revendications 1 à 6, dans lequel ledit dispositif est conçu pour mémoriser des données dans une pluralité de pistes et lesdites régions de positionnement sont des régions d'asservissement incluant des informations de positionnement de tête préécrites définissant les positions desdites pistes et dans lesquelles au moins une région de données est séparée par une région d'asservissement.

9. Dispositif de stockage de données selon l'une quelconque des revendications précédentes, dans lequel ledit dispositif est conçu pour mémoriser des données dans une pluralité de pistes, chaque piste incluant une indication de début de piste.

10. Dispositif de stockage de données selon l'une quelconque des revendications précédentes, dans lequel ledit moyen de tête inclut une tête pour écrire des informations sur ledit disque et une tête magnéto-résistive pour lire les informations à partir dudit disque.

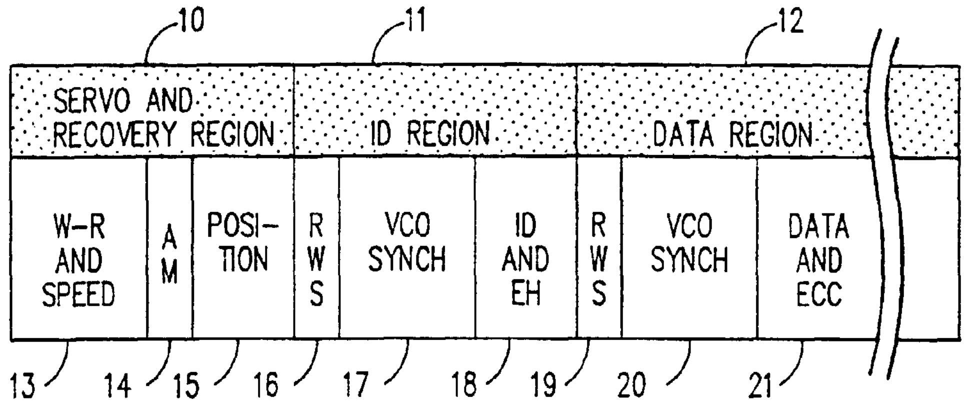

FIG. 1A: PRIOR ART SECTOR SERVO SECTOR FORMAT

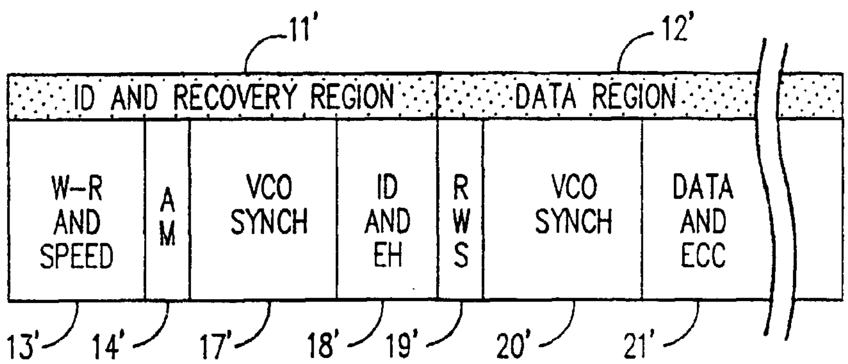

FIG. 1B: PRIOR ART DEDICATED SERVO SECTOR FORMAT

Bad Sector List							
Physical Sector	3	5	6	9	10	11	13
Logical to Physical Mapping							
Logical Sector	1	2	3	4	5	6	7
Physical Sector	1	2	4	7	8	12	14
Look-up Table							
Logical Sector	3	4	6	7			
Offset	1	3	6	7			

FIG. 3: EXAMPLE OF LOOK-UP TABLE CREATION

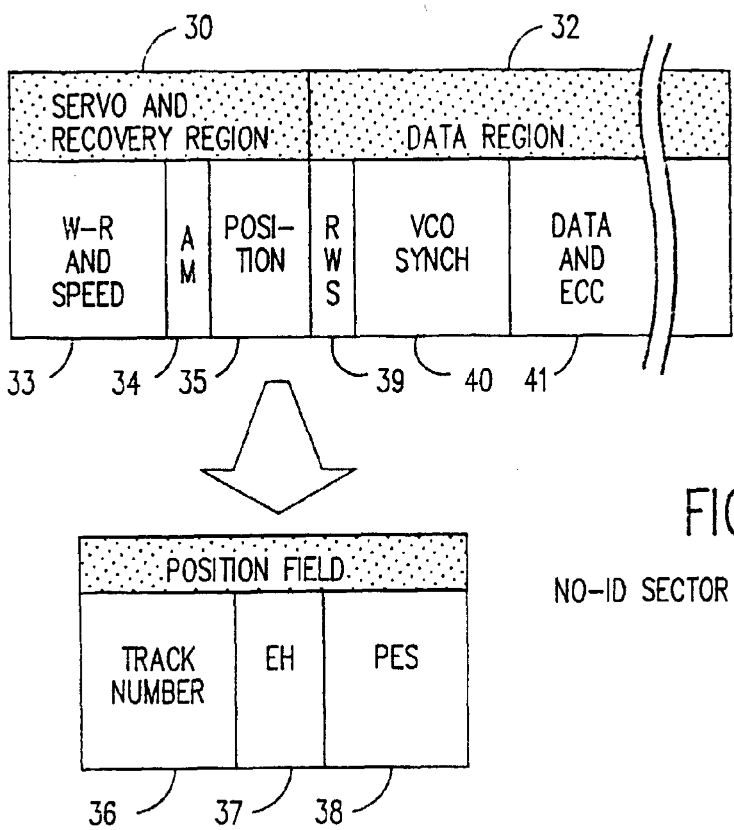

FIG. 2A:

NO-ID SECTOR SERVO SECTOR FORMAT

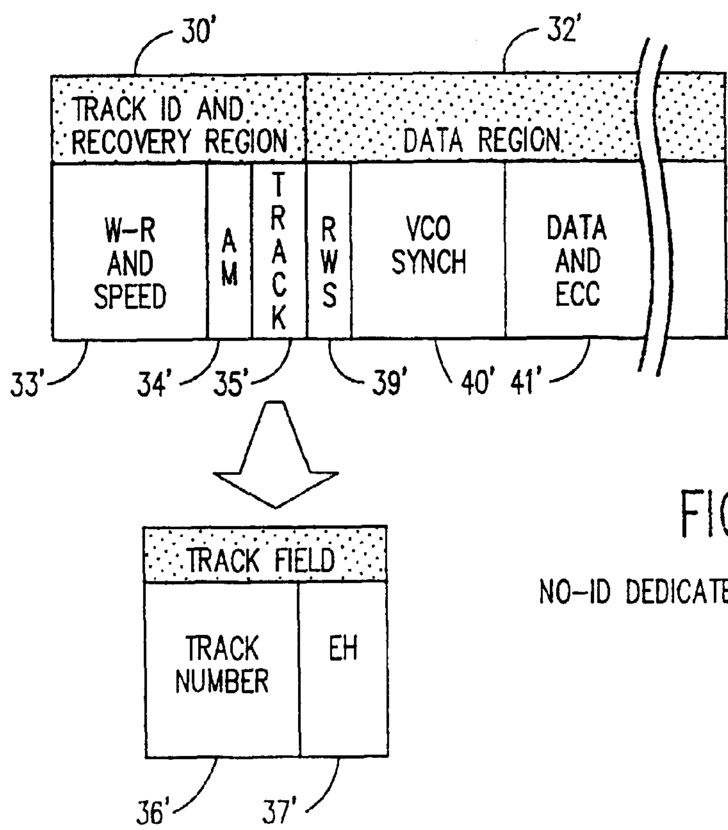

FIG. 2B:

NO-ID DEDICATED SERVO SECTOR FORMAT

```
static int tl[MAXBAD];                  /*lookup table: logical #*/
static int to[MAXBAD];                  /*lookup table: offset*/
static int nt = 0;                      /*number table entries*/

void build(b,nbad)                      /*build bad sector table*/
int b[MAXBAD];                          /*physical bad sectors*/
int nbad;                               /*number of bad sectors found*/
{
  int bad;

  bad = 0;                              /*bad sector counter*/
  nt = 0;                               /*table index counter*/

  tl[nt] = b[bad];                      /*first entry*/
  to[nt] = 1;                           /*offset starts at 1*/
  bad++;                                /*next bad sector*/
  while (b[bad] == b[bad-1]+1) {        /*while adjacent sectors*/
    to[nt]++;                           /*increment offset*/
    bad++;                              /*next sector*/
}

 while (bad < nbad) {
   nt++;                                /*next table entry*/
   tl[nt] = b[bad] - to[nt-1];          /*will be next entry*/
   to[nt] = to[nt-1] + 1;               /*starting offset*/
   bad++;                               /*next bad sector*/
   if (bad == nbad) break;              /*end of the list*/
   while (b[bad] == b[bad-1]+1) {       /*while adjacent sectors*/
     to[nt]++;                          /*increment offset*/
     bad++;                             /*next sector*/
     if (bad == nbad) break;            /*end of the list*/
   }
 }
 nt++;                                  /*nt is now # elements*/
 if ((nt & 1) == 0) {                   /*table should be odd*/
   tl[nt] = 32767;                      /*max allowed value*/
   nt++;
 }
} /* build */
```

FIG. 4: Look-up Table Generation Algorithm in C.

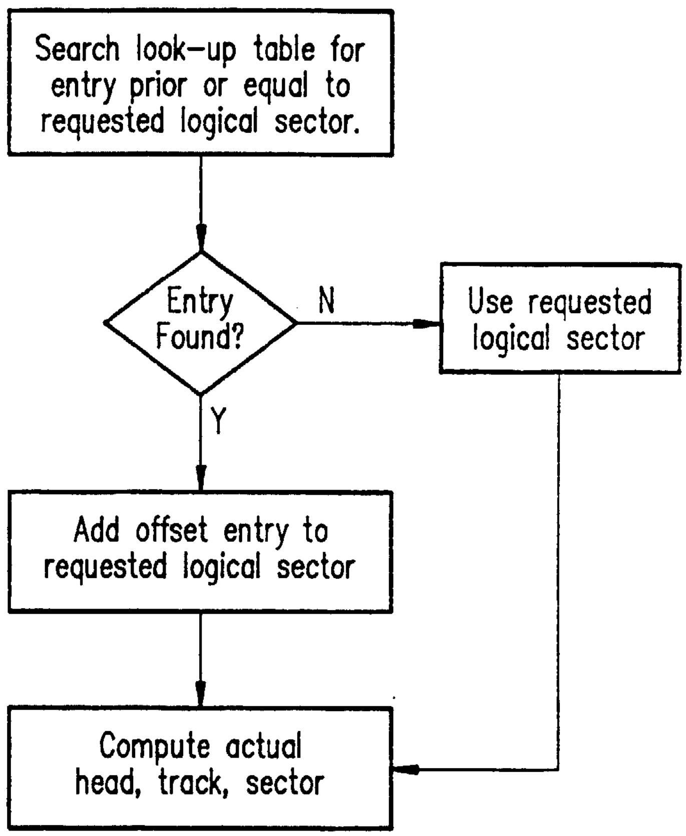

FIG. 5: Logical-to-Physical Sector Conversion Algorithm