9700286-T
SP

A DATA RECORDING DISK FILE WITH FIXED BLOCK ARCHITECTURE SECTOR THAT ELIMINATES THE IDENTIFICATION (ID) REGION.

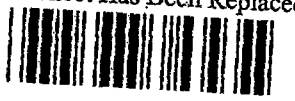
This invention relates to A multiple track data storage disk, a method of formatting and a data recording disk file.

Background of the Invention

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.

A typical FBA sector servo sector format according to the prior art is illustrated in Fig. 1A. The sector is divided into three regions: servo and recovery region 10, ID region11 and data region12. Generally, servo and recovery region 10 contains overhead associated with

This Sheet Has Been Replaced



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.

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.

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

This Sheet Has Been Replaced



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 he 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.

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.

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.

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

This Sheet Has Been Replaced

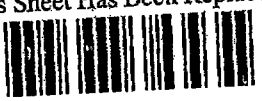


SUMMARY OF THE INVENTION

According to the present invention there is disclosed a data recording disk having a pattern of radially spaced tracks and a plurality of circumferentially spaced angular sectors having prerecorded servo positioning information for identifying track and sector locations, each of the plurality of sectors including only a data region for recording user data and, preceding the data region in the circumferential direction, a servo-and-recovery region which provides the prerecorded servo positioning information to uniquely identify the sector and track, wherein the servo-and-recovery region includes a write-to-read-recovery and speed-compensation field, an error handling field, a position-error-signal field, a start-of-sector indicating mark, and a track number field, but no data region identification (ID) field, wherein the data region includes, in sequence, a read-to-write-recovery and speed-compensation field and a synchronization field, and wherein one of the plurality of sectors also includes a start-of-track indicating mark.

According to another aspect of the present invention there is disclosed a data recording disk having a plurality of tracks, each of the plurality of tracks having a start-of-track indicating mark and being divided into a number of data regions for recording user data and a number of servo-and-recovery regions, the servo-and-recovery regions containing a position-error-signal field, a servo-and-recovery region count mark, and a track number field, at least one of the data regions being split by a servo-and-recovery region, and wherein none of the plurality of tracks have, either within or outside the data regions and servo-and-recovery regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks.

This Sheet Has Been Replaced



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

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

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

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

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

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.

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

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

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

This Sheet Has Been Replaced



37 and PES subfield 38. Data region 32 includes R-W and speed field 39, VCO synch field 40 and data field 41.

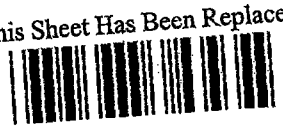
In operation, fields 33, 34, 39, 40 and 41 provide substantially the same functions as fields 13, 14, 19, 20 and 21 shown in prior art Fig. 1A and described previously. Fields 16, 17 and 18 from prior art Fig. 1A are eliminated.

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.

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.

The no-id sector format of the present invention provides several space saving advantages over the prior art. The track number need not

This Sheet Has Been Replaced



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.

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.

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

This Sheet Has Been Replaced



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

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.

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.

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

This Sheet Has Been Replaced



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 surface, and thus may provide an attractive alternative that is within the scope and spirit of the present invention.

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

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.

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.

This Sheet Has Been Replaced



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.

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

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.

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

This Sheet Has Been Replaced



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.

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.

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).

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,

This Sheet Has Been Replaced



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.

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.

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.

In some implementations (e.g. SCSI), it may be desirable to allow for the relocation of bad sectors without resorting to a complete device

This Sheet Has Been Replaced



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.

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.

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

This Sheet Has Been Replaced



No-ID format of the present invention eliminates this problem since it has no ID field to be read.

This Sheet Has Been Replaced



CLAIMS:

1. A data recording disk having a pattern of radially spaced tracks and a plurality of circumferentially spaced angular sectors having prerecorded servo positioning information for identifying track and sector locations, each of the plurality of sectors including only a data region for recording user data and, preceding the data region in the circumferential direction, a servo-and-recovery region which provides the prerecorded servo positioning information to uniquely identify the sector and track, wherein the servo-and-recovery region includes a write-to-read-recovery and speed-compensation field, an error handling field, a position-error-signal field, a start-of-sector indicating mark, and a track number field, but no data region identification (ID) field, wherein the data region includes, in sequence, a read-to-write-recovery and speed-compensation field and a synchronization field, and wherein one of the plurality of sectors also includes a start-of-track indicating mark.

2. The data recording disk as recited in claim 1, wherein the track number field includes a full track number prerecorded therein.

3. The data recording disk as recited in claim 1, wherein the start-of-track indicating mark is encoded in the track number field.

4. The data recording disk as recited in claim 1, wherein the start-of-sector indicating mark is an address mark located in the servo-and-recovery region, and wherein the address mark precedes the track number field in the circumferential direction.

5. The data recording disk as recited in claim 1, wherein the track number field includes a full sector number prerecorded therein.

6. The data recording disk as recited in claim 1 wherein some of the tracks also have servo-and-recovery regions located within some of the data regions.

7. A data recording disk having a plurality of tracks, each of the plurality of tracks having a start-of-track indicating mark and being divided into a number of data regions for

* C L M S *

This Sheet Has Been Replaced

recording user data and a number of servo-and-recovery regions, the servo-and-recovery regions containing a position-error-signal field, a servo-and-recovery region count mark, and a track number field, at least one of the data regions being split by a servo-and-recovery region, and wherein none of the plurality of tracks have, either within or outside the data regions and servo-and-recovery regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks.

8. The data recording disk as recited in claim 7 wherein the track number field includes a full track number prerecorded therein.

9. The data recording disk as recited in claim 7 wherein the start-of-track indicating mark is located within one of the servo-and-recovery regions.

10. The data recording disk as recited in claim 9 wherein the start-of-track indicating mark is encoded in the track number field.

11. The data recording disk as recited in claim 7 wherein the servo-and-recovery region count mark is a start-of-servo-and-recovery region mark.

12. The data recording disk as recited in claim 11 wherein the start-of-servo-and-recovery region mark is an address mark that precedes the track number field in the track.

13. The data recording disk as recited in claim 11 wherein the start-of-servo-and-recovery region mark is a servo-and-recovery region number.

14. The data recording disk as recited in claim 7 wherein the servo-and-recovery regions include a write-to-read recovery and speed-compensation field.

15. The data recording disk as recited in claim 7 wherein the data regions include, in sequence, a read-to-write recovery and speed-compensation field and a synchronization field.



This Sheet Has Been Replaced



A DATA RECORDING DISK FILE WITH FIXED BLOCK ARCHITECTURE SECTOR THAT ELIMINATES THE IDENTIFICATION (ID) REGION.

ABSTRACT

A data recording disk file with a fixed block architecture sector format that eliminates the ID region. The servo region contains sector identification information in the form of a start-of-track indicating mark recorded in a selected sector of each track and a start-of-sector indicating mark recorded in each sector. A full track number identifier is encoded in the position field within the servo region of each sector. A look-up table is built at format time to map bad sectors out of the disk file. The table contains entries for each cluster of contiguous bad sectors, and is searched at runtime to provide conversion of logical sector location identifiers to physical sector location identifiers.

Fig. 2A

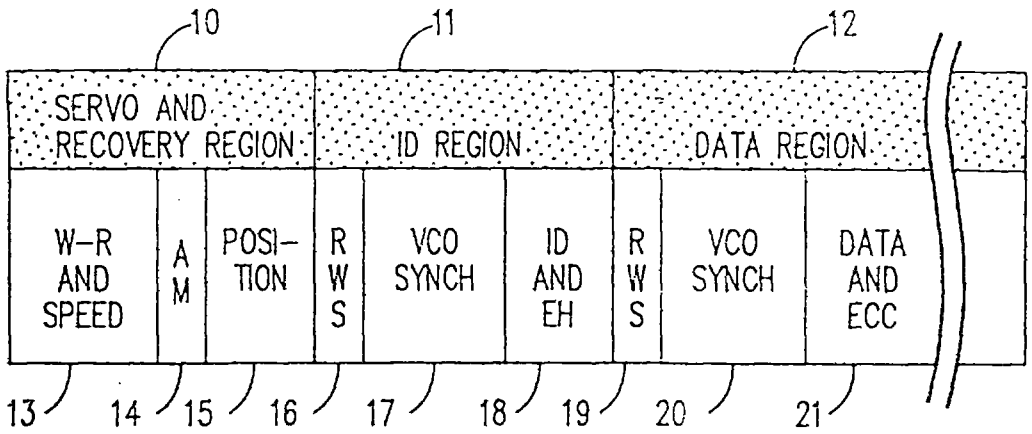


FIG. 1A: PRIOR ART SECTOR SERVO SECTOR FORMAT

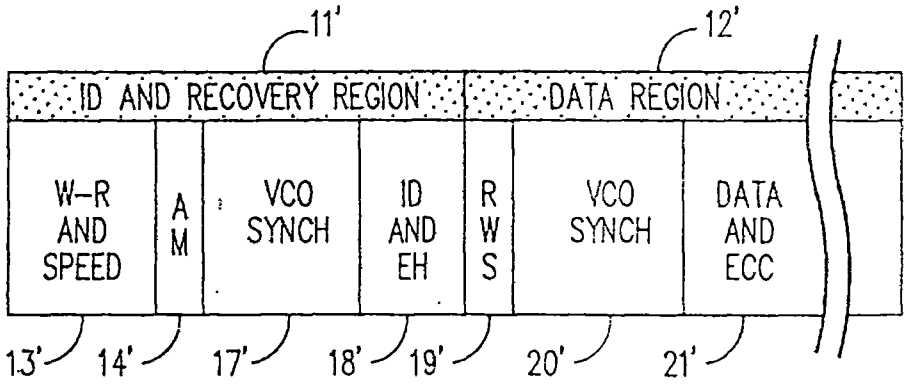


FIG. 1B: PRIOR ART DEDICATED SERVO SECTOR FORMAT

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

This Sheet Has Been Replaced

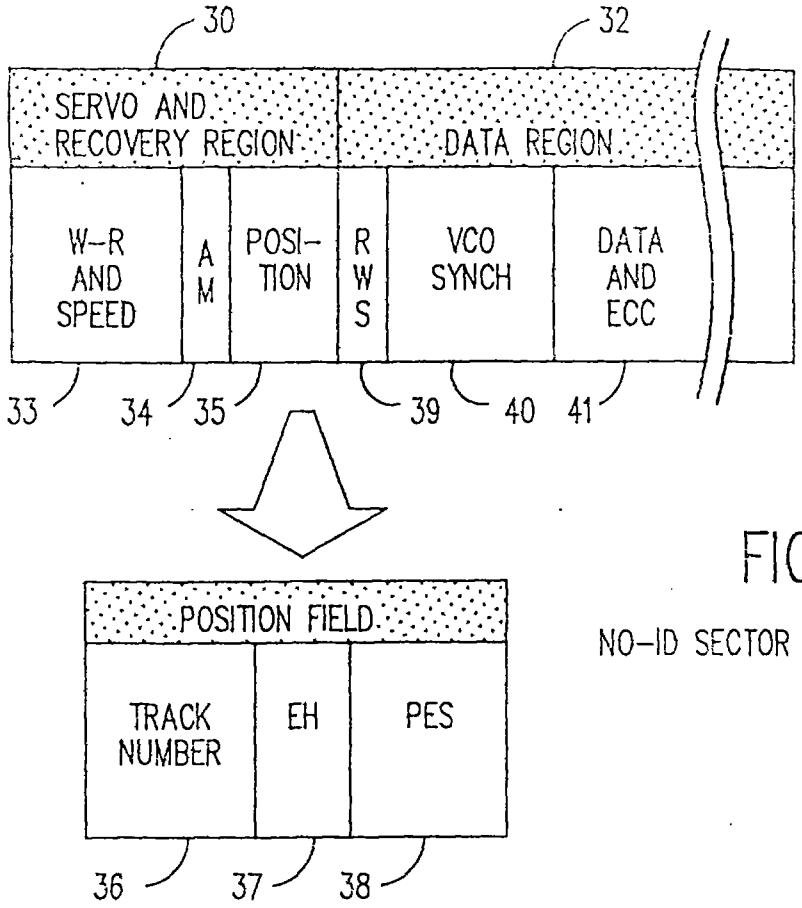


FIG. 2A:

NO-ID SECTOR SERVO SECTOR FORMAT

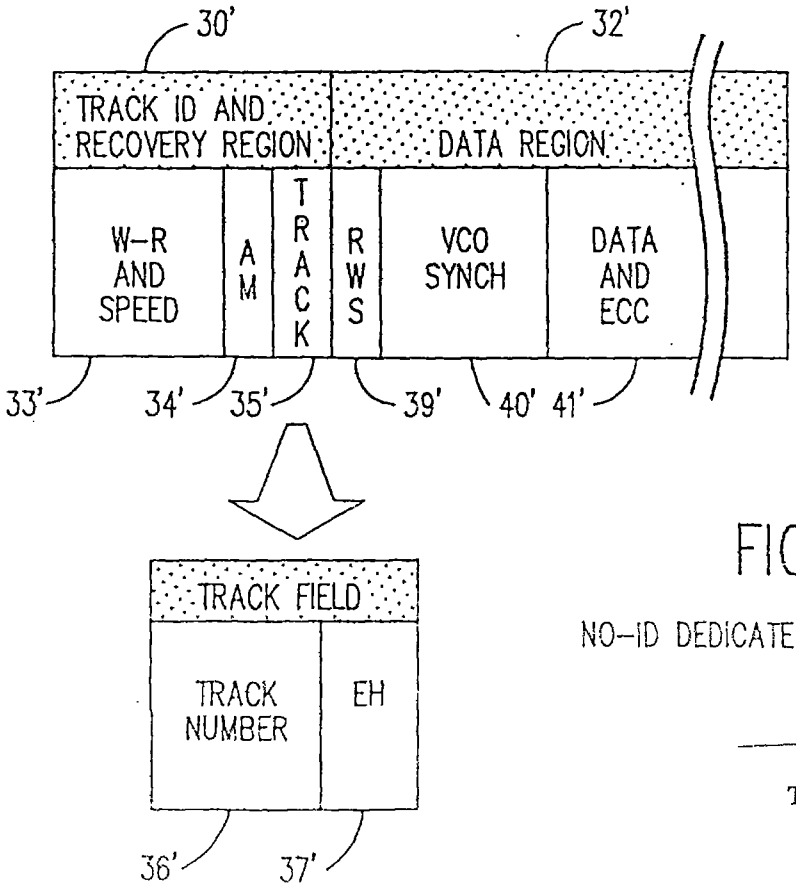


FIG. 2B:

NO-ID DEDICATED SERVO SECTOR FORMAT

This Sheet Has Been Replaced



```
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 */
```

This Sheet Has Been Replaced



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

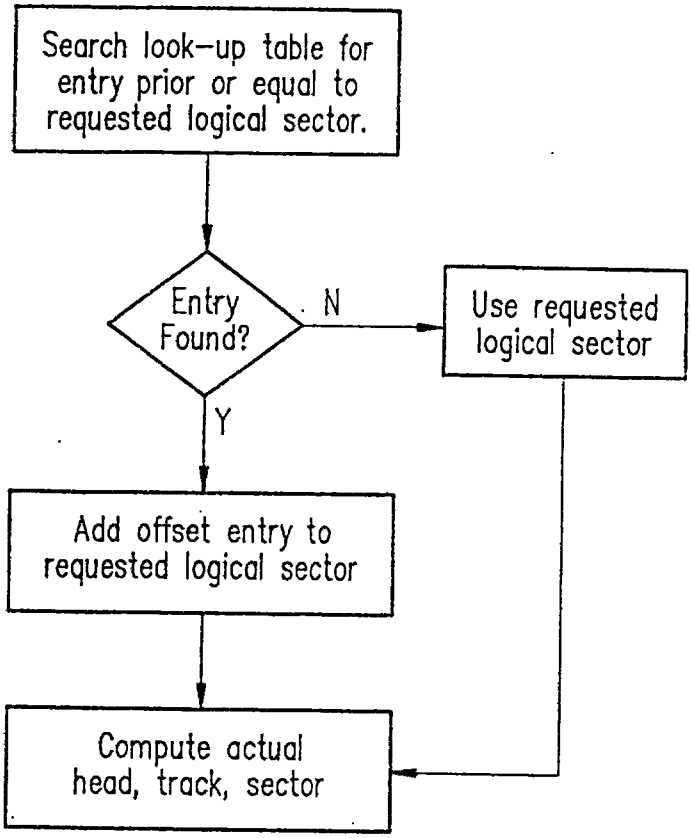


FIG. 5: Logical-to-Physical Sector Conversion Algorithm

This Sheet Has Been Replaced

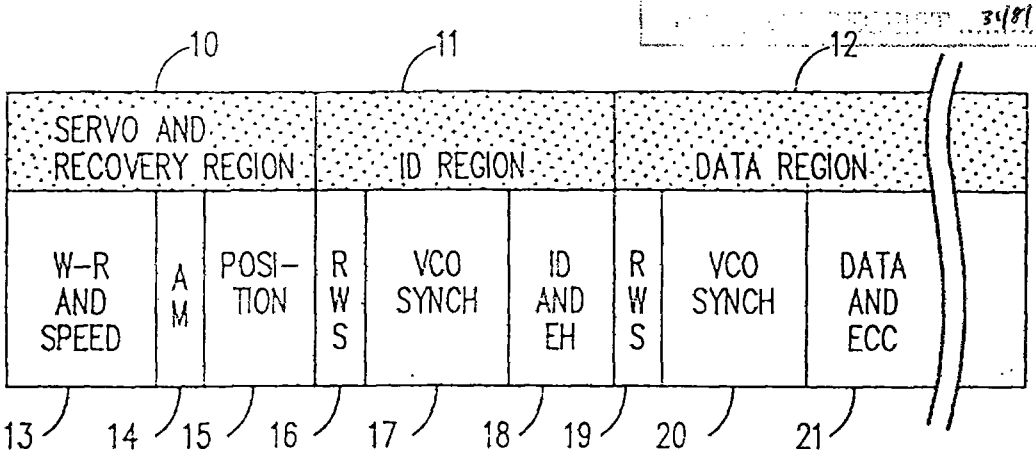


FIG. 1A

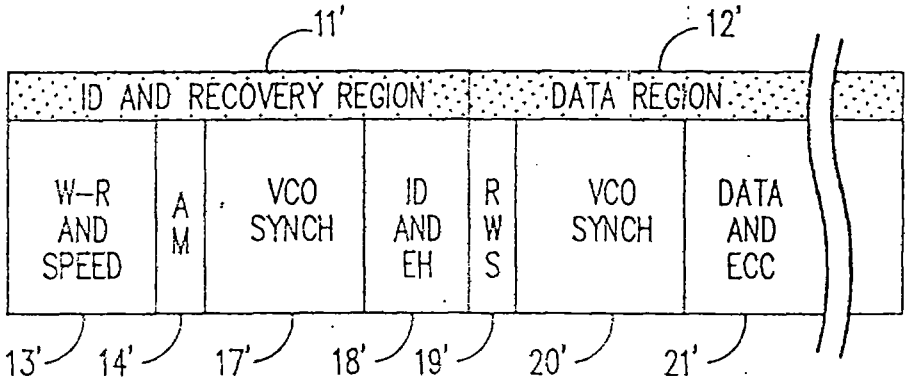


FIG. 1B

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

REPLACEMENT SHEET
31 AUG 1998

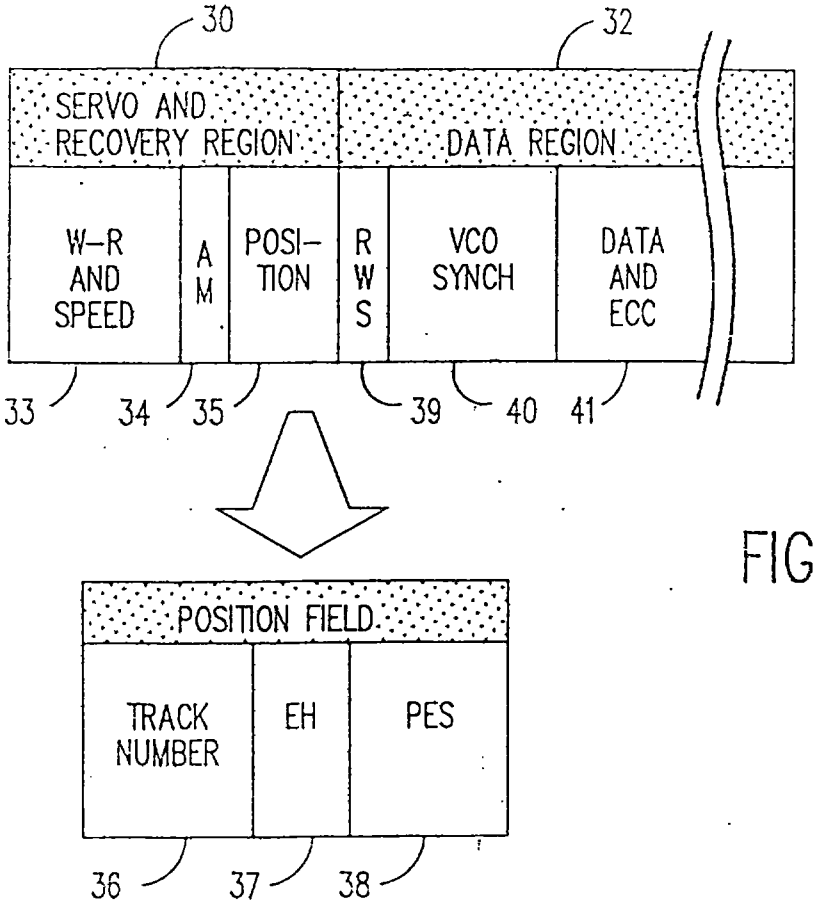


FIG. 2A

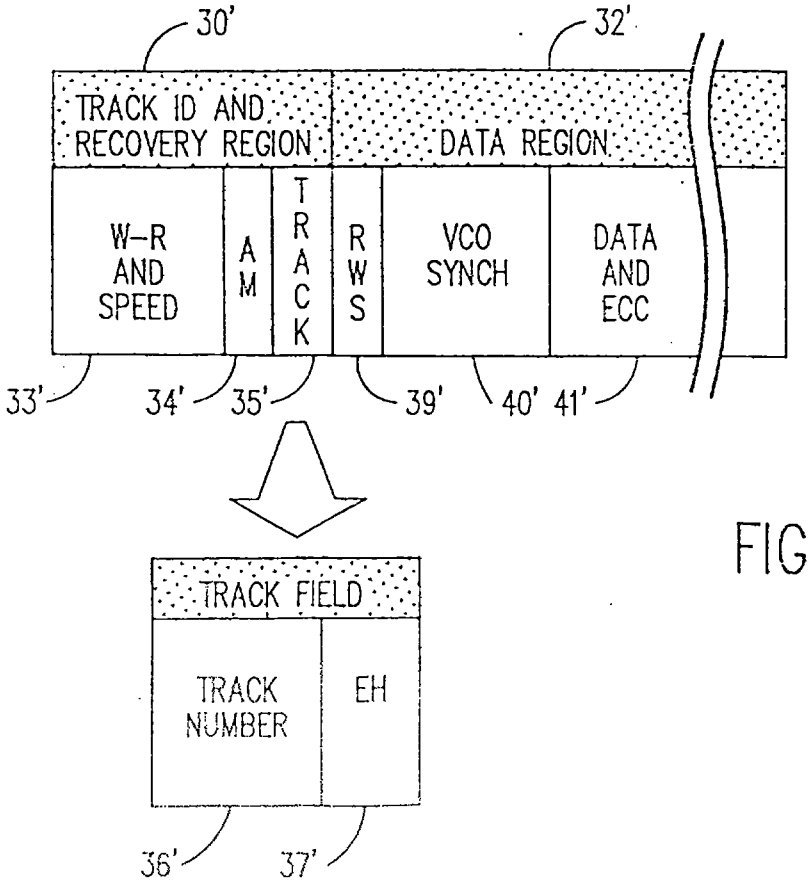


FIG. 2B

REPLACEMENT SHEET
31 AUG 1992

```
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

REPLACEMENT SHEET
31 AUG 1998

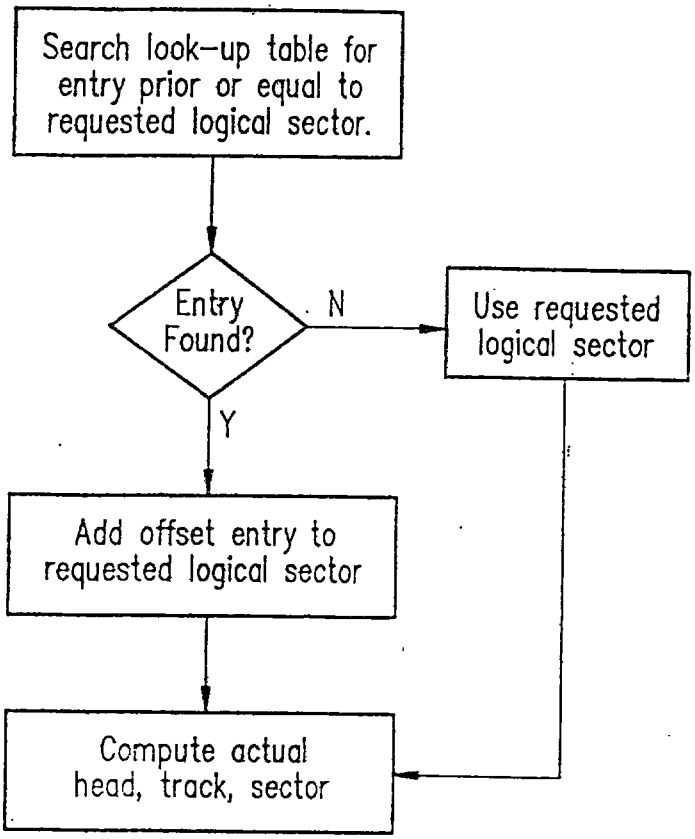


FIG. 5

31 AUG 1998

9706328-8 IBC3

16/11/98

A DATA RECORDING DISK FILE WITH FIXED BLOCK ARCHITECTURE SECTOR THAT ELIMINATES THE IDENTIFICATION (ID) REGION.

This invention relates to A multiple track data storage disk, a method of formatting and a data recording disk file.

Background of the Invention

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.

A typical FBA sector servo sector format according to the prior art is illustrated in Fig. 1A. The sector is divided into three regions: servo and recovery region 10, ID region11 and data region12. Generally, servo and recovery region 10 contains overhead associated with

This Sheet Has Been Replaced

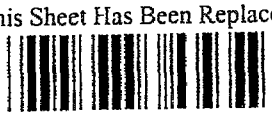


sectoring, as well as servo information for sector servo files. Also, this region marks the beginning of the sector. Specifically, servo region 10 contains WRITE-to-READ recovery (W-R) and speed compensation field 13, which is used to switch the data channel from WRITE to READ and accommodate spindle 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.

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.

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

This Sheet Has Been Replaced



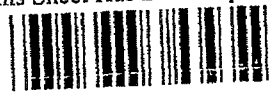
ID region requires a great deal of disk space to perform its functions. Since the disk uses the information in the ID region to perform a logical-to-physical sector conversion in order to locate he physical position on the disk corresponding to the requested logical sector, the ID region must be large 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.

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.

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.

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

This Sheet Has Been Replaced



SUMMARY OF THE INVENTION

According to the present invention there is disclosed a data recording disk having a pattern of radially spaced tracks and a plurality of circumferentially spaced angular sectors having prerecorded servo positioning information for identifying track and sector locations, each of the plurality of sectors including only a data region for recording user data and, preceding the data region in the circumferential direction, a servo-and-recovery region which provides the prerecorded servo positioning information to uniquely identify the sector and track, wherein the servo-and-recovery region includes a write-to-read-recovery and speed-compensation field, an error handling field, a position-error-signal field, a start-of-sector indicating mark, and a track number field, but no data region identification (ID) field, wherein the data region includes, in sequence, a read-to-write-recovery and speed-compensation field and a synchronization field, and wherein one of the plurality of sectors also includes a start-of-track indicating mark.

According to another aspect of the present invention there is disclosed a data recording disk having a plurality of tracks, each of the plurality of tracks having a start-of-track indicating mark and being divided into a number of data regions for recording user data and a number of servo-and-recovery regions, the servo-and-recovery regions containing a position-error-signal field, a servo-and-recovery region count mark, and a track number field, at least one of the data regions being split by a servo-and-recovery region, and wherein none of the plurality of tracks have, either within or outside the data regions and servo-and-recovery regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks.

This Sheet Has Been Replaced



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

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

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

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

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

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.

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

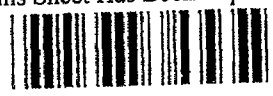
Fig. 5 is a flowchart illustrating an algorithm for performing logical-to-physical sector conversion according to the present invention.

Detailed Description of the Invention

IA. NO-ID FBA SECTOR SERVO SECTOR FORMAT

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

This Sheet Has Been Replaced



37 and PES subfield 38. Data region 32 includes R-W and speed field 39, VCO synch field 40 and data field 41.

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.

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.

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.

The no-id sector format of the present invention provides several space saving advantages over the prior art. The track number need not

This Sheet Has Been Replaced

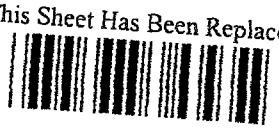


be redundantly encoded in an ID region but instead is communicated to the DDC by the servo system at runtime. Track number subfield 36 and EH subfield 37 are substantially shorter than ID region 11 shown in prior art Fig. 1A, since the FBA format of the present invention is able to use CRC rather than 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.

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.

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

This Sheet Has Been Replaced



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.

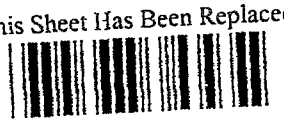
1B. NO-ID FBA DEDICATED SERVO SECTOR FORMAT

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.

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.

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

This Sheet Has Been Replaced



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 surface, and thus may provide an attractive alternative that is within the scope and spirit of the present invention.

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

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.

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.

This Sheet Has Been Replaced



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.

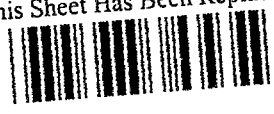
There are a number of variations on the embodiment that serve to improve error tolerance where AM field 34 is missed due to a disk defect. First, the physical sector number may be encoded into track number field 36. Second, an open loop timer may be used to provide a timeout error if an address mark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

III. LOOK-UP TABLE

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.

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

This Sheet Has Been Replaced



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.

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.

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).

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,

This Sheet Has Been Replaced



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.

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.

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.

In some implementations (e.g. SCSI), it may be desirable to allow for the relocation of bad sectors without resorting to a complete device

This Sheet Has Been Replaced

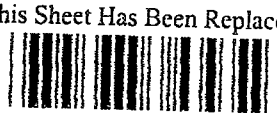


format operation. This feature is implemented in conjunction with the present invention by allocating a set of sectors as spares prior to an initial format operation. The spares are treated as though they are bad sectors for purposes of constructing the logical-to-physical mapping table. The spare sectors 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.

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.

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

This Sheet Has Been Replaced



No-ID format of the present invention eliminates this problem since it has no ID field to be read.

This Sheet Has Been Replaced



TT00358Q IPC/
✓ 4/4a —
✓ 15. 17.
RECEIPT 22/12/99

SUMMARY OF THE INVENTION:

It is an object of the present invention to ameliorate one or more disadvantages of the prior art.

According to one aspect of the present invention there is provided fixed-block architecture embedded servo disk drive comprising: a data recording disk having a plurality of tracks, each track being divided into a number of data regions and servo regions, the servo regions containing servo region count information, and wherein none of the tracks have, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks; a recording head which reads information in the servo regions and which writes and reads user data in preselectedones of the data regions; and a disk data controller coupled to the recording head for identifying the preselected data regions based on servo region count information read by the recording head from circumferentially preceding servo regions without use of ID regions.

According to another aspect of the present invention there is provided a fixed-block architecture embedded servo disk drive comprising: a data recording disk having a plurality of tracks, each track being divided into a number of data regions and servo regions, the servo regions containing servo region count information, at least one of the data regions being split by a servo region, and wherein none of the tracks have, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks; a read head adjustable to different radial positions and capable of reading information in the servo regions, including a servo region that splits a data region; a write head different from the read head for writing user data in the data regions, including a split data region; and a disk data controller for initiating read and write operations, a write operation of user data in a split data region being preceded by reading of servo region count information from servo regions, including a servo region that splits a data region, but not preceded by reading of information from an ID region.

According to still another aspect of the present invention there is provided a fixed-block architecture embedded servo disk drive comprising: a data recording disk having a plurality of tracks, each track being divided into a number of data regions, at least one of the data regions containing a defect where data cannot be written, and servo regions containing servo region count information, and wherein none of the tracks have, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks; a recording head which reads information in the servo regions and which, writes and reads data in the data regions; a defect map which contains information relating to the locations of

REPLACEMENT SHEET
27 DEC 1999
REPLACED SHEETS SG
✓ Description : paged 4/4a to paged —
☐ Claim(s) : paged ..... to paged .....
☐ Drawing(s) : paged ..... to paged .....
☐ Abstract : paged ..... to paged .....
DATE OF RECEIPT .....
This Sheet Has Been Replaced

defects on the disk; and a disk data controller coupled to the recording head for locating the data regions without defects where data is to be written by use of the servo region count information in the circumferentially preceding servo regions read by the recording head in combination with information read from the defect map but without use of ID regions.

According to still another aspect of the present invention there is provided a fixed-block architecture embedded servo disk drive comprising: a data recording disk having a plurality of tracks, each track having a start-of-track indicating mark and being divided into data regions and servo regions containing position fields and servo region count information, none of the tracks having, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks; a magnetoresistive read head which reads position fields and servo region count information in the servo regions and user data in the data regions; a write head not radially aligned with the read head and which writes user data in the data regions; and data control electronics which identify the data regions where data is to be written based on detection of the start-of-track indicating mark and reading of servo region count information in the circumferentially preceding servo regions by the read head; whereby user data is written in the data regions by the write head without radially jogging the write head following reading of the preceding servo regions by the read head but without reading of information from ID regions.

This Sheet Has Been Replaced



REPLACEMENT SHEET

22 DEC 1998

CLAIMS

Having thus described the invention, what is claimed is:

1. A fixed-block architecture embedded servo disk drive comprising:

a data recording disk having a plurality of tracks, each track being divided into a number of data regions and servo regions, the servo regions containing servo region count information, and wherein none of the tracks have, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks;

a recording head which reads information in the servo regions and which writes and reads user data in preselectedones of the data regions; and

a disk data controller coupled to the recording head for identifying the preselected data regions based on servo region count information read by the recording head from circumferentially preceding servo regions without use of ID regions.

2. The disk drive as recited in claim 1, wherein at least one of the data regions is split by a servo region.

3. The disk drive as recited in claim 1, wherein the servo region count information is servo region number information.

4. The disk drive as recited in claim 1, each of the tracks further including a start-of-track indicating mark, the data regions being identified based on the start-of-track indicating mark and a count of the number of servo regions.

5. The disk drive as recited in claim 1, further including a timer, responsive to detection of a servo region, which provides a timeout signal indicating failure to detect an expected servo region.

6. The disk drive as recited in claim 1, wherein the servo regions include error handling fields.

7. The disk drive as recited in claim 1, wherein the servo regions include position fields having track numbers recorded therein.

8. The disk drive as recited in claim 7, wherein the track numbers are full track numbers.

9. The disk drive as recited in claim 1, further including a defect map, the data regions being identified based on information from the defect map.

10. A fixed-block architecture embedded servo disk drive comprising:

a data recording disk having a plurality of tracks, each track being divided into a number of data regions and servo regions, the servo regions containing servo region count information, at least one of the data regions being split by a servo region, and wherein none of the tracks have, either within or outside the data regions and servo

REPLACEMENT SHEET
22 DEC 1998
* C L M S *

regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks;

a read head adjustable to different radial positions and capable of reading information in the servo regions, including a servo region that splits a data region;

a write head different from the read head for writing user data in the data regions, including a split data region; and

a disk data controller for initiating read and write operations, a write operation of user data in a split data region being preceded by reading of servo region count information from servo regions, including a servo region that splits a data region, but not preceded by reading of information from an ID region.

11. The disk drive as recited in claim 10, each of the tracks further including a start-of-track indicating mark, the split data region where data is to be written being located by detecting the start-of-track indicating mark and counting servo regions.

12. The disk drive as recited in claim 10, wherein the servo regions include error handling fields.

13. A fixed-block architecture embedded servo disk drive comprising:

a data recording disk having a plurality of tracks, each track being divided into a number of data regions, at least one of the data regions containing a defect where data cannot be written, and servo regions containing servo region count information, and wherein none of the tracks have, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks;

a recording head which reads information in the servo regions and which, writes and reads data in the data regions;

a defect map which contains information relating to the locations of defects on the disk; and

a disk data controller coupled to the recording head for locating the data regions without defects where data is to be written by use of the servo region count information in the circumferentially preceding servo regions read by the recording head in combination with information read from the defect map but without use of ID regions.

14. The disk drive as recited in claim 13, wherein the defect map information converts logical sector numbers to physical sector numbers.

15. The disk drive as recited in claim 13, wherein the data regions without defects are located by data sector numbers derived from the servo region count information and the defect map.

16. The disk drive as recited in claim 15, each of the tracks further including a start-of-track indicating mark and the servo region count information including start-of-servo-

* C L M S *

REPLACEMENT SHEET

22 DEC 1998

sector marks, the data regions without defects being located by detecting the start-of-track indicating mark and counting start-of-servo-sector marks.

17. The disk drive as recited in claim 13, wherein at least one of the data regions is split by a servo region.

18. The disk drive as recited in claim 13, wherein the servo regions include error handling fields.

19. A fixed-block architecture embedded servo disk drive comprising:

a data recording disk having a plurality of tracks, each track having a start-of-track indicating mark and being divided into data regions and servo regions containing position fields and servo region count information, none of the tracks having, either within or outside the data regions and servo regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks;

a magnetoresistive read head which reads position fields and servo region count information in the servo regions and user data in the data regions;

a write head not radially aligned with the read head and which writes user data in the data regions; and

data control electronics which identify the data regions where data is to be written based on detection of the start-of-track indicating mark and reading of servo region count information in the circumferentially preceding servo regions by the read head; whereby user data is written in the data regions by the write head without radially jogging the write head following reading of the preceding servo regions by the read head but without reading of information from ID regions.

20. The disk drive as recited in claim 19, wherein the servo regions have track number information recorded therein and the data control electronics are responsive to the track number information.

21. The disk drive as recited in claim 10, wherein the read head is a magnetoresistive read head.

22. The disk drive as recited in claim 19, wherein the start-of-track indicating mark is an index mark.

23. The disk drive according to claim 19, wherein the start-of-track indicating mark is located in one of the servo regions.

24. The disk drive as recited in claim 19, wherein the servo region count information is a servo sector number.

25. The disk drive as recited in claim 19, wherein the servo region count information is a start-of-servo-sector mark.

26. The disk drive as recited in claim 19, wherein at least one of the data regions is split by a servo region.

REPLACEMENT SHEET
22 DEC 1999
* C L M S *

A DATA RECORDING DISK FILE WITH FIXED BLOCK ARCHITECTURE SECTOR THAT ELIMINATES THE IDENTIFICATION (ID) REGION.

This invention relates to A multiple track data storage disk, a method of formatting and a data recording disk file.

Background of the Invention

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.

A typical FBA sector servo sector format according to the prior art is illustrated in Fig. 1A. The sector is divided into three regions:

REPLACEMENT SHEET
10 MAY 1999

DATE OF RECEIPT 10/3/99

servo and recovery region 10, ID region 11 and data region 12. 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 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.

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 VGO 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.

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

REPLACEMENT SHEET
10 MAR 1993

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 mis-identifying 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 eel accommodate spindle speed variations.

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.

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.

REPLACEMENT SHEET
10 MAR 1989

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

Summary of the Invention

According to the present invention there is disclosed a data recording disk having a pattern of radially spaced tracks and a plurality of circumferentially spaced angular sectors having prerecorded servo positioning information for identifying track and sector locations, each of the plurality of sectors including only a data region for recording user data and, preceding the data region in the circumferential direction, a servo-and-recovery region which provides the prerecorded servo positioning information to uniquely identify the sector and track, wherein the servo-and-recovery region includes a write-to-read-recovery and speed-compensation field, an error handling field, a position-error-signal field, a start-of-sector indicating mark, and a track number field, but no data region identification (ID) field, wherein the data region includes, in sequence, a read-to-write-recovery and speed-compensation field and a synchronization field, and wherein one of the plurality of sectors also includes a start-of-track indicating mark.

According to another aspect of the present invention there is disclosed a data recording disk having a plurality of tracks, each of the plurality of tracks having a start-of-track indicating mark and being divided into a number of data regions for recording user data and a number of servo-and-recovery regions, the servo-and-recovery regions containing a position-error-signal field, a servo-and-recovery region count mark, and a track number field, at least one of the data regions being split by a servo-and-recovery region, and wherein none of the plurality of tracks have, either within or outside the data regions and servo-and-recovery regions, identification (ID) regions containing information that uniquely identifies the data regions in the tracks.

REPLACEMENT SHEET
10 MAR

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

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

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

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

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

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.

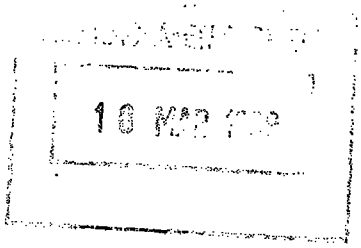
Fig. 4 is a C language procedure illustrating an algorithm for creating a look-up table according to the present invention.

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

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.

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.

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.

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 or 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.

REPLACEMENT SHEET

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.

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.

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

REPLACEMENT SHEET
10 MAR 1993

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.

lB. NO-ID FBA Dedicated Servo Sector Format

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.

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. lB and described previously. Fields 17' and 18' from prior art Fig lB are eliminated. Track number subfield 36' and EH subfield 37' are identical to fields 36 end 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.

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

REPLACEMENT SHEET
10 MAY 1993

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 overheads since field 35' is not placed on the data surface, and thus may provide an attractive alternative that is within the scope and spirit of the present invention.

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

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.

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

REPLACEMENT SHEET
10 MAR 1995

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.

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.

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

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.

REPLACEMENT SHEET
10 MAR 1999

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.

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.

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).

REPLACEMENT SHEET
10 MAR 1993

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 (ie 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.

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.

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

REPLACEMENT SHEET
19 MAR 1999

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.

In some implementations (eg. 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.

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.

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-

ENCLOSURE SHEET
10 MAY 1990

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.

REPLACEMENT SHEET
10 MAR 1999