

5

METHOD OF OVERFLOW-BASED DEFECT MANAGEMENT REPRESENTATION

Related Applications

The present application claims benefit of the United States
10 provisional patent application number 60/205,928, filed May 22, 2000.

Field of the Invention

The present invention relates generally to disc drives, and in
15 particular to media defect management schemes.

Background of the Invention

Defect management schemes in a disc drive identify segments of the
20 disc surface that are defective and which cannot be used to store data. Such
defects may be caused by a number of sources, including thermal asperity,
dust particles or other contaminants. These defective locations need to be
identified and coded as no useful information can be recorded at these
defective sites.

25 Drive certification is performed as part of the manufacturing process
to identify defects on the media. In a typical full slip defect management
scheme, all defective sectors found are characterized by a slip count
parameter. The drive electronics uses this parameter to skip over the
defective sectors. The accumulated slip and the cylinder location are
30 recorded in the defect slippage field and the cylinder field respectively.
Both these fields are a single word size parameter. This implies that the
largest number of accumulated slips that can be recorded is 65535. This
limitation poses a potential problem as the capacity of disc drives increases
and the potential number of slips may exceed this number.

5 If a track or cylinder is mapped out due to bad servo bursts, the accumulated slip count increases tremendously, and the slip count may exceed the limit.

 Following the current trend in the disc drive industry to increase drive capacity by increasing the number of tracks per inch, a similar
10 problem will arise when the number of cylinders in a drive exceed 65535, since the cylinder field is a single word parameter.

 A natural solution is to use an additional memory byte or word to represent the slip count or cylinder count, and thus avoid any possibility of parameter overflow. Unfortunately, this is not a feasible solution as buffer
15 memory is both scarce and expensive in the disc drive electronics system. For example, if an additional byte is added for the cylinder and slip count fields, the additional memory required for the entire defect table will be 2 bytes multiplied by the number of defect entries. For an 8K defect entry system, an additional 16K bytes will be required.

20 There is clearly an unmet need for some way to overcome this memory limitation at acceptable cost.

Summary of the Invention

25 In accordance with the invention, there is provided a method for storing values, some of the values being wider than n bits. The method is especially useful in a disc drive having n -bit wide memory allocated to the storage of such values.

 According to a most preferred embodiment, the method involves
30 obtaining the difference between each of the values and a value $k2^n$. The value k is chosen as the smallest integer for the difference to be less than 2^n . The differences are stored in a n -bit wide first memory in order of increasing magnitude of the values. Each of the differences is referenced by an address. For each k value, the address of the smallest difference is stored

5 in a second memory as an overflow address. The method may further include updating the addresses in the first memory and, if necessary, in the second memory, when a new difference is inserted into the first memory. In this manner, the invention allows for the storage of values that may be wider than n bits without resorting to allocating more than n bits to store
10 each of the values.

The method is applicable whether or not the values are accumulative. If the values are accumulative in nature, the n-bit values are updated along with any required updating of the addresses when a new difference is inserted into the first table.

15 To retrieve a value, the address of the difference stored in the first memory is compared with the overflow addresses stored in a second memory. The value of 2^n is added to the difference if an overflow address is equal to or greater than the address. The comparing and adding steps are repeated for each overflow address less than the address of the difference.
20 The final sum is the value to be retrieved.

Thus, it can be understood that the present invention can be implemented with minor code changes and yet bring about significant increase in the magnitude of the values that can be stored. These and various other features as well as advantages, which characterize the
25 present invention, will be apparent from the following detailed description.

Brief Description of the Drawings

FIG. 1 shows an exemplary disc drive.

30 FIG. 2 provides a schematic illustration of defective sectors in a track.

FIG. 3 is a flowchart showing the storage of a number according to a preferred embodiment of the present invention.

5 FIG. 4 is a flowchart of a preferred method of reading back the stored number.

Detailed Description

10 A preferred embodiment of the present invention will be described with the aid of the attached drawings. Shown in FIG. 1 is an exemplary disc drive such as one in which the present invention may be implemented. The disc drive 10 includes at least one disc 12 to which data can be recorded and retrieved from by read/write heads 14 supported by an actuator
15 assembly 16. Servo electronics 18 operably connected to the actuator assembly controls the position of the read/write heads 14.

 The disc is divided into tracks 20. Tracks which are the same distance from the center of the discs are referenced by the same cylinder number. A head number is used for designating the different disc surfaces.
20 Each track 20 is further divided into a number of sectors 22, each sector being an addressable location for data storage. The physical address of a sector may therefore be expressed in the Cylinder/Head/Sector (CHS) format.

 A defect location is represented in a defect table by its cylinder,
25 head, sector and its span. The defect span can be obtained by subtracting the previous defect entry slip count from the current slip count. Generally, a sector is also identified by a logical address. While the physical address refers to the actual location of the sector on the disc, the logical address is the address used by an associated host system to read data from and write
30 data to the disc drive. When a sector is found to be not suitable for use in storing data, its logical address is assigned to another sector so that the defective sector will not be utilized for storing data. In this manner, the host system is presented with an apparently contiguous series of memory storage units.

5 Various schemes have been devised to skip over defective sectors,
whether identified at the time of drive certification or found to have
become defective in the course of use. Typically such schemes involve
writing data to each sector and reading back the data to determine if the
sector is suitable for use in storing data. If the sector is found to be
10 defective, the logical address of the sector is assigned to the next good
sector. It is therefore possible to build a defect table that identifies defective
sectors by their physical addresses and the corresponding slip counts. The
slip count is an accumulative counter that serves to indicate the relative
difference between the logical address and the physical address.

15 The defect table is usually stored in a reserved part of the disc.
When the disc drive is powered up for operation, the defect table is
transferred to the buffer memory. There, it is accessed by the drive
electronics to direct the disc drive to skip over the defective sectors during
operation. Generally, the cylinder numbers, slip counts and indices in the
20 following examples are expressed in hexadecimal, and it is assumed that
the memory space allocated to the storage of each cylinder number or slip
count is 16 bits.

 Table 1 below represents a defect table in which a new entry
representing defective sectors is inserted. Supposing that in this example
25 the new entry represents defects at Cylinder = FF2E, Head = 0, Sector = 34
with Span = 5 as shown by the shaded boxes 23 in FIG. 2. This new entry
will be inserted at Index = 1013 so that the defect entries remain in
ascending order of the CHS address. The new defect entry appears as
Cylinder = FF2E, Head = 0, Sector = 34, Slip = (FFFD + 5) = 10002. Making
30 reference to step 24 in FIG. 3, the Slip value is split into a 16-bit value of
0002 and an overflow value of 1. The value that is actually stored in the
defect table is the 16-bit value as shown in step 24. The address of this
defect entry, which is given by the Index, is stored in a second table as
shown in Table 2, here referred to as an overflow table, unless the overflow

- 5 table already stores a smaller or equal overflow Index value for the same overflow value, as shown in step 28.

Table 1

	Index	Cylinder	Head	Sector	Slip
10	1010	EE20	0	23	FFEC
	1011	FA20	1	34	FFF2
	1012	FD34	0	5A	FFFD
	1013	FF2E	0	34	0002

15

Table 2

Overflow Index

1013

- 20 The Slip counts of all the defect entries, if any, below the new defect entry are updated by the addition of the Span associated with the new defect entry, as shown in step 30. For example, if Table 3 represents the defect entries in the defect table before the new defect entry of Cylinder = FF2E, Head = 0, Sector = 34 with Span = 5 is inserted, Table 4 shows the state of the defect table after the insertion and updating is completed.

25

Table 3

	Index	Cylinder	Head	Sector	Slip
	1010	EE20	0	23	FFEC
	1011	FA20	1	34	FFF2
	1012	FD34	0	5A	FFFD
30	1013	FFE2	0	2F	001A

The defect entry that used to have an address of Index = 1013, and now addressed by Index = 1014, has its Slip count increased from 1001A to $(1001A + 5) = 1001F$. This new Slip count is similarly split into a 16-bit

5 value of 001F and an overflow value of 1. The actual value stored in the defect table is 001F. Of all the defect entries that are associated with the same overflow value, the address of the defect entry having the smallest Index value is that which is stored in the overflow table. In this example, the first entry in the overflow table remains as 1013.

10

Table 4

	Index	Cylinder	Head	Sector	Slip
	1010	EE20	0	23	FFEC
	1011	FA20	1	34	FFF2
15	1012	FD34	0	5A	FFFD
	1013	FF2E	0	34	0002
	1014	FFE2	0	2F	001F

In another embodiment where the actual Slip count exceeds 1FFFF
20 and the memory allocated to the defect table for the storage of the Slip value remains at 16 bits, the overflow table will have two entries. The first being the address of the first defect entry having a Slip count greater than FFFF, and the second being the address of the first defect entry having a Slip count greater than 1FFFF. This can be extrapolated to store very large
25 numbers. For example, to store the number 30104, the 16-bit value 0104 is obtained and stored in the defect table. The overflow value = 3. If this defect entry is the first defect entry associated with the same overflow value of 3, the address of this defect entry will be stored in the overflow table. If the overflow table contains an existing overflow index for the same
30 overflow value, and the existing overflow index points to a larger address than the new defect entry with the same overflow value, the address of the new defect entry will replace the existing overflow index in the overflow table, as in step 32. If the existing overflow index in the overflow table

5 points to an address that is smaller or equal to that of the new entry, for the same overflow value, the existing overflow index is retained.

To retrieve the actual slip count of a defect entry, its address is compared with the overflow address stored in the overflow table. One preferred embodiment is illustrated by steps 34, 36 and 38 of a flowchart in
10 FIG. 4. Referring back to the example of Table 1, to retrieve the Slip count of the defect entry stored at Index = 1013, the Index value 1013 is compared with each overflow index in the overflow table, until the Index value is found to be smaller than the overflow index. For every overflow index that is less than or equal to the Index value, the Slip value is incremented by
15 10000. In this example, the actual Slip count of 1001A is thus obtained.

It is envisaged that the number of cylinders may one day exceed FFFF. The present invention provides a way to circumvent a memory limitation problem that may potentially arise in that situation. In one embodiment of the present invention, there is provided another overflow
20 table, here referred to as the cylinder overflow table, that is configured to store a cylinder overflow index or cylinder overflow indices.

Consider, for example, the insertion of a defect entry having Cylinder = 10213, Head = 1, Sector = 12E, Span = DF at Index = 1015 in a defect table represented by Table 5.

25

Table 5

	Index	Cylinder	Head	Sector	Slip
	1010	EE20	0	23	FFEC
	1011	FA20	1	34	FFF2
30	1012	FD34	0	5A	FFFD
	1013	FF2E	0	34	0002
	1014	FFE2	0	2F	001F
	1015	0213	1	12E	00FE

5 The Cylinder number is split into a 16-bit value of 0213 and a cylinder overflow of 1. The 16-bit value is stored in the defect table. The address of the defect entry Index = 1015 is stored in the cylinder overflow table since this is the first defect entry with a Cylinder number larger than FFFF. Table 6 shows the cylinder overflow table for this case.

10

Table 6
Cylinder Overflow Index
1015

15 The Slip count of this defect entry = $DF + 1001F = 100FE$. The value of 00FE is stored in the defect table. The slip count overflow index that is stored in a slip count overflow table is not changed since the first defect entry to have a Slip count exceed FFFF remains as that associated with Index = 1013.

20 If another defect entry having Cylinder = 1012E, Head = 1, Sector = 0, Span = 4, is inserted into the defect table, it will be inserted at Index = 1015. The defect entry previously at Index = 1015 will acquire a new Index of 1016, as shown in Table 7. The resultant defect table has 012E stored under Cylinder at Index = 1015, and 0213 stored under Cylinder at Index =
25 1016. The Slip count of the defect entry now at Index = 1016 is updated from 00FE to 0102 after adding the Span of 4 introduced by the defect entry at Index = 1015.

Table 7

30

Index	Cylinder	Head	Sector	Slip
1010	EE20	0	23	FFEC
1011	FA20	1	34	FFF2
1012	FD34	0	5A	FFFD
1013	FF2E	0	34	0002

5	1014	FFE2	0	2F	001F
	1015	012E	1	0	0023
	1016	0213	1	12E	0102

To retrieve the actual Cylinder number for a defect entry at any
10 given Index, the Cylinder number stored in the defect table is increased by
10000 for every cylinder overflow index found to be less than or equal to
the Index.

To support a maximum of 256K cylinders or slip counts, an
overflow table with 3 words of memory to store 3 overflow indices is
15 needed. In general, the largest number (cylinder number or slip counts)
that can be supported by a system of the present invention is given by
 $((m+1)(w+1)-1)$, where m is the largest number that can be stored in the
defect table alone and where w is the number of words allocated to the
overflow table.

20 It has already been shown that the invention is flexible, and allows
for the storage of numbers like cylinder numbers and accumulative
numbers like slip counts. It is to be understood that other numbers may be
stored efficiently in generally the same manner, with slight modifications
to accommodate any special characteristics of the numbers. In other words,
25 only minor code changes are required for the present invention to be
applied to existing storage systems. This is an advantage as a complete
redesign of the entire defect table will not be required. Neither is there a
need for the buffer memory to be increased.

Another situation which may be encountered is described below
30 with reference to Table 8 which represents a defect table and Table 9 which
represents an overflow table for the Slip count.

Table 8

Index	Cylinder	Head	Sector	Slip
-------	----------	------	--------	------

5	1011	FA20	1	24	FFF2
	1012	FA20	1	5A	FFFD
	1013	FA20	1	6B	0004

Table 9

10	Slip Overflow Index
	1013

If a new defect entry of Cylinder = FA20, Head = 1, Sector = 65 with Span = 6 is inserted into the defect table, the defect entries at Index = 1012 and Index = 1013 will be merged, with the new entry, into a single defect entry at Index = 1012, giving the defect table of Table 10. The overflow address in the overflow table, in this example, becomes 1012, as shown in Table 11, because after the merger, the defect entry having the smallest difference for the overflow value of 1 is the defect entry having an address of Index = 1012.

Table 10

	Index	Cylinder	Head	Sector	Slip
	1011	FA20	1	24	FFF2
25	1012	FA20	1	5A	000A

Table 11

	Slip Overflow Index
	1012

30

Tables 13 represents a Slip overflow table before a new defect entry of Cylinder = EE20, Head = 0, Sector = 40, Span = 4, is inserted into the defect table represented by Table 12.

5

Table 12

Index	Cylinder	Head	Sector	Slip
1010	EE20	0	23	FFEC
1011	EE20	0	34	FFF3
1012	EE20	0	44	0000

10

Table 13

Slip Overflow Index
1012

15

The new defect entry can be merged with the defect entry at Index = 1012, giving the entries as shown in Table 14.

Table 14

Index	Cylinder	Head	Sector	Slip
1010	EE20	0	23	FFEC
1011	EE20	0	34	FFF3
1012	EE20	0	40	0004

20

No change is required in the overflow table since the defect entry with the smallest difference remains as that with an address of Index = 1012.

25

The present invention may be further described as follows:

A method of the present invention for storing a value includes a step 24 of obtaining the difference between the value and a value $k2^n$ if the value is greater than 2^n-1 . The value k is chosen as the smallest integer for the difference to be less than 2^n . The differences are stored in a n -bit wide first memory in order of increasing magnitude of the values 26. Each of the differences is referenced by an address. For each k value, the address of the

30

5 smallest difference is stored in a second memory as an overflow address
28.

 The method may further include providing a first address to a first
difference. This first difference is obtained from a first value. New
addresses are assigned to the differences obtained from numbers greater
10 than the first value. In one embodiment, for each k value, the address of the
smallest difference stored in the second memory as an overflow address
may be replaced with the new address of the smallest difference 32. The
value may be a cylinder number.

 In an alternative embodiment, the method may include obtaining an
15 increment value by subtracting the first value from a second value. The
second value is selected from the largest value that is less than the first
value. In one embodiment, for each difference obtained from a value
greater than the first value, a new difference is obtained by adding the
increment value to the difference. The new differences are then stored at
20 the respective new addresses 30.

 For each new difference that is equal to or greater than 2^n , a third
difference between the new difference and a value $k2^n$ is obtained. The
value k is chosen as the smallest integer for the third difference to be less
than 2^n . In one embodiment, the third differences are stored at the
25 respective new addresses 30, and for each k value, the new address of the
smallest third difference is stored as overflow address in the second
memory 32. The value may be an accumulative slip count.

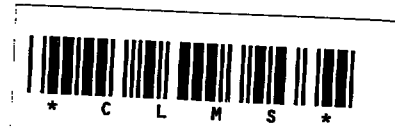
 To retrieve the value, the address of the difference stored in a n -bit
wide first memory is compared with an overflow address stored in a
30 second memory 36. The value of 2^n is added to the difference if the address
is equal to or greater than the overflow address 38. This is repeated for
each overflow address less than the address 36.

 While the invention has been described with reference to particular
preferred embodiments, it will be appreciated by those skilled in the art

5 that changes may be made with respect to form or detail without departing
from the scope of the present invention. For example, the memory
allocated to the storage of the number of interest has been described as
being 16 bits long. The invention is just as applicable if the user has defined
the memory allocation to be n bits long, where n is an integer. Where n bits
10 is the allocated memory space in a main table, the largest number that can
be stored by n bits alone will be $2^n - 1$. Any number greater than that will be
split into an n -bit value to be stored in the main table and an overflow
value of at least 1.

5

Claims



1. In a disc drive having a disc formatted with a plurality of tracks, a read/write head positionable relative to the disc for reading from and
10 writing to the disc, and circuitry for performing a method of storing values, the method comprising steps of:
 - (a) for each value greater than 2^n-1 , obtaining the difference between the value and a value $k2^n$, wherein k is chosen as the smallest integer for the difference to be less than 2^n ;
 - 15 (b) storing the differences in a n -bit wide first memory in order of increasing magnitude of the values, wherein each of the differences is referenced by an address; and
 - (c) for each k value, storing the address of the smallest difference in a second memory as an overflow address.
- 20 2. The method of Claim 1 wherein the storing step (b) further includes steps of:
 - (d) providing a first address to a first difference, the first difference being obtained from a first value; and
 - 25 (e) assigning new addresses to the differences obtained from numbers greater than the first value.
3. The method of Claim 2 further comprising, for each k value, a step
30 (f) of replacing the overflow address with the new address of the smallest difference.
4. The method of Claim 2 wherein the value is a cylinder number.



- 5 5. The method of Claim 2 wherein the storing step (b) further includes steps of:
- (g) obtaining an increment value by subtracting the first value from a second value, the second value being the largest value that is less than the first value;
- 10 (h) for each difference obtained from a value greater than the first value, obtaining a new difference by adding the increment value to the difference; and
- (i) storing the new differences at the respective new addresses.
- 15 6. The method of Claim 5 further comprising steps of:
- (j) for each new difference that is equal to or greater than 2^n , obtaining a third difference between the new difference and a value $k2^n$, wherein k is chosen as the smallest integer for the third difference to be less than 2^n ;
- 20 (k) storing the third differences at the respective new addresses; and
- (l) for each k value, storing the new address of the smallest third difference in the second memory as the overflow address.
7. The method of Claim 5 wherein the value is an accumulative slip
- 25 count.
8. In a disc drive having a disc formatted with a plurality of tracks, a read/write head positionable relative to the disc for reading from and writing to the disc, and circuitry for performing a method of retrieving a
- 30 value, the method comprising steps of:
- (a) comparing an address of a difference from a n -bit wide first memory with an overflow address stored in a second memory; and
- (b) adding 2^n to the difference if the address is equal to or greater than the overflow address.



5

9. The method of Claim 8 further comprising a step (c) of repeating the adding step (b) for each overflow address that is less than the address.

10. A disc drive comprising:

10

a n-bit wide memory; and

means for storing values greater than 2^n-1 .