



RTMP's copy

## METHOD FOR REDUCING ROTATIONAL LATENCY IN A DISC DRIVE

The present invention relates to seek operations in rotating media and in particular in rotating disc drive memory assemblies. More particularly, the present invention relates to a method for sorting the command queue to reduce rotational latency in disc drive memory assemblies.

Rotating disc memories include one or more discs driven about a vertical spindle axis. Each disc has a plurality of concentric tracks on one or both surfaces from which information is read or onto which information is written by a read/write element. In a disc drive system, a disc rotates at a high speed while the read/write element "flies" over the surface of the rotating disc. The read/write element is positioned over specific areas of the disc in accordance with commands received from the host computer.

The SCSI interface protocol provides a mechanism for host computers to send multiple commands to a disc drive. It allows the disc drive to determine the optimum order of execution for the commands. The set of commands available for execution by the disc drive is referred to as the "command queue".

Traditionally, disc drives have sorted the command queue in an order that minimizes the seek time between the various commands. However, seek time is only one of two components of the true access time of a command. The other component is the rotational latency time or the amount of time the disc drive spends waiting for the appropriate data to rotate under the read/write element. The rotational latency time is a significant part of the total access time for all but the longest seeks. On average, it is the dominant component of the total access time for relatively short seeks.

To determine the optimum order of execution of a set of commands, it is desirable to consider the full access time between commands rather than simply the seek time. An ideal sorting algorithm would sort the command queue in such a way that the sum of the access times of all of the commands in the queue is minimized. This would involve computing the sum of the access times for each of the possible orders of the



queue. Computing access times for every possible order, however, is far too computationally intensive to do in a real-time system such as a disc drive. For example, there are over three million possible orders in a set of ten commands. Modern disc drives can have up to 64 or more commands in the command queue. In addition, the command queue in practical applications is dynamic. In other words, new commands are constantly received from the host computer. Thus, the optimum order of execution of previously received commands often changes, requiring the entire queue to be re-sorted.

Since the above-mentioned ideal algorithm is impractical, an alternative method of sorting is required. One improved method of sorting would be to choose the next command to be executed by searching the command queue to find the command with the shortest access time. Although this does not result in a guaranteed minimum sum of all command access times, it comes relatively close.

To compute the access time of an arbitrary command, the disc drive controller firmware must first compute the seek time for the command. The seek time is the time required to move the read/write element from the present track to the desired track. Next, it must determine the expected rotational position of the disc drive after the seek is complete. Since different disc drives will have different seek characteristics, the firmware must provide for some margin of error when computing seek times. For very long seeks, this margin of error may be quite large and hence, impractical for implementation of the present invention. A sorting algorithm which minimizes rotational latency for commands with relatively short seeks, however, would be a significant contribution to the art.

In a first aspect, this invention provides, in a system including a data retrieval element and a rotating medium, a method for decreasing rotational latency characterised by comprising;

building a seek range array corresponding to discrete angular regions on the medium;

searching a command queue array for a command in a selected discrete angular region; and



executing the command found in the selected discrete angular region of the medium.

In a second aspect, this invention provides, in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek characterised by comprising:

- dividing the disc into discrete angular regions;
- building a seek range array corresponding to the discrete angular regions;
- building a command queue array;
- searching the command queue array for a command in successive discrete angular regions; and
- executing the command found in a discrete angular region.

In a third aspect, this invention provides, in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek characterised by comprising:

- building a seek range array;
- searching a command queue array for a command;
- computing a rotational latency from a current position of the read/write element to the command in the command queue;
- executing the command found in the command queue.

In a fourth aspect, this invention provides a system including a rotating medium having a plurality of tracks, a transducing element adjacent the rotating element and a command queue array containing commands identifying locations on tracks, a process of moving the transducing element from adjacent a location on a first track to adjacent a location on a different track identified by a command with minimal rotational latency characterised by comprising:

- building a seek range array defining locations on plural tracks that may be accessed within one revolution of the medium as the transducing element is moved from the first track;



selecting a location in the seek range array identified by a command in the command queue array; and

moving the transducing element to the selected location.

In a fifth aspect, this invention provides in a disc drive system including a transducing element and a rotating media, a process to decrease access time during a short seek characterised by comprising:

building a seek range array defining locations on plural tracks that may be accessed within one revolution of the media as the transducing element is moved from the first track;

identifying commands in a command queue array addressing locations in the seek range array;

computing rotational latencies from a current location of the transducing element to locations addressed by identified commands in the command queue array; and

selecting the identified command having the smallest rotational latency.

The present invention is a method to minimize rotational latency for short seeks. More particularly, the method requires the firmware to search the command queue for commands which have a starting logical block address (LBA) that is 0.1 to 0.2 of a revolution away from the current rotational position of the read/write element. If one of these commands has a starting cylinder number which is close enough to the current cylinder position of the read/write element to allow a seek to complete within 0.1 of a revolution, then that particular command is executed. Otherwise, the firmware searches the command queue for commands with a starting LBA that is within 0.2 to 0.3 of a revolution away from the current position of the read/write element. If a command can be completed within a seek time of 0.2 of a revolution or less, then that particular command will be executed. If no such command is available, the firmware searches for commands with a starting LBA that is 0.3 to 0.4 of a revolution away, etc. If no commands are found which can be accessed within one revolution of the disc, the firmware searches the command queue for the command which has the shortest expected seek time.



The firmware must have some method of determining the difference in rotational position between two LBAs. Although this could be accomplished by converting the LBA numbers into a physical address (i.e. cylinder, head, sector) the sector numbers of LBAs in different recording zones cannot be compared easily. To resolve this situation, the method requires the firmware to divide the disc into a number (e.g. 100) of discrete angular regions. For the purpose of sorting the queue, the starting and ending LBA for each command is converted into a cylinder, head, and angular region address. The angular region numbers (e.g.  $N=1 \dots N=100$ ) provide a common system of measurement for all LBAs regardless of recording zone. These angular region numbers are always treated "modulo 100". For example, the difference in rotational position from angular region 90 to angular region 5 is 15. The firmware treats an angular region difference of 10 to 19 as 0.1 of a revolution and 20 to 29 as 0.2 of a revolution, etc.

Embodiments of the invention will now be described by way of example only, with reference to the accompanying diagrammatic figures in which;

Figure 1 shows a plan view of a rotating disc, portions of the disc which can be reached within one revolution of the current position of the read/write element are shaded;

Figure 2 shows the rotating disc of Figure 1 divided into discrete angular regions as required by the method of the present invention; and

Figure 3 is a flow diagram of the steps of the method of the present invention and is split between Figures 3A and 3B, identical parts having the same reference numerals throughout.

Figure 1 shows a plan view of rotating disc 10. Read/write element 12 is shown positioned at location  $X_1$  on disc 10. Shaded area 14 represents the seek range [N] or the number of cylinders that can be traversed by read/write element 12 during one disc revolution from its current position at point  $X_1$ . For example, assume that read/write element 12 is able to cross a total of 200 cylinders in one revolution of disc 10. More particularly, within one revolution, starting from point  $X_1$ , read/write element 12 will be able to traverse 200 cylinders. At point  $X_1$ , shaded region 14 is very small because a

**THIS SHEET HAS BEEN  
REPLACED**

very small percentage of the disc revolution is complete. At point  $X_2$ , however, half of a revolution has passed. The number of cylinders that read/write element 12 can traverse within half a disc revolution is approximately 80-120 cylinders. Correspondingly, the width of shaded area 14 at point  $X_2$  has significantly increased with respect to the amount of shaded area at point  $X_1$ . At point  $X_3$ , an entire disc revolution is almost complete. Accordingly, the portion of shaded area 14 immediately adjacent point  $X_3$  represents the greatest area of the disc that may be crossed in one revolution (approximately 200 cylinders).

Figure 2 shows rotating disc 10 and shaded area 14 of Figure 1 modelled in discrete angular regions as required by the method of the present invention. More particularly, disc 10 is divided into ten angular regions labelled  $N=0$ ,  $N=1$ ,  $N=2$ ,  $N=3$ ... $N=9$  wherein "N" is relative to the current actuator position.

In operation, read/write element 12 may be positioned at point  $Y_1$  which is the boundary between regions  $N=0$  and  $N=9$ . The method requires the firmware to initially search the angular region labelled  $N=1$  for an executable command. If a command is found within region  $N=1$ , the firmware then determines if the number of cylinders read/write element 12 must traverse to reach the start of the found command is within the seek range for that particular angular region. If the command is within the seek range (i.e. the shaded area), the command is executed. If no command is found within the seek range, the firmware searches angular region  $N=2$  for a command within the seek range of that region.

Each angular region is searched in a clockwise fashion until an entire revolution of the disc is complete. If no commands can be found within any of the seek ranges (i.e. shaded areas) of each region, the firmware executes the command which requires the shortest seek time with respect to the current position of the read/write element.

Figure 3 is a flow diagram of the steps of the method of the present invention. In step 30, a seek range array is built. The seek range array is indexed according to time in units of tenths of a revolution. Each element of the array contains the number of cylinders which can be crossed in the amount of time corresponding to the element's



index. For example, seek range [3]=100 means that a seek can cross 100 cylinders in 3/10 of a revolution.

In step 32, the firmware determines if the command queue is empty. As long as the queue is empty, no action will be taken. Note, however, that commands may be added to the command queue at any time. In step 34, it has been determined that the command queue is not empty, so the firmware targets angular region  $N=1$ . In step 36, the firmware accesses the first command (i.e. index=1) of the command queue index. The command queue index is an array that contains each command in the command queue. Thus, command queue [1] is the first command in the queue, command queue [2] is the second command, etc.

In step 38, the firmware computes the rotational latency from the current position of the read/write element to the command in command queue [1].

In step 40, the firmware determines if the rotational latency computed in step 38 is  $N/10$ ths to  $(N+1)/10$ ths of a revolution away from the current position of the read/write element (i.e. within the parameters of region  $N=1$ ). If the rotational latency falls within the parameters which define angular region  $N=1$ , in step 42, the "cylinder delta" is computed. The cylinder delta is the number of cylinders from the current position of the read/write element to the start cylinder of the command in command queue [1].

In step 44, the firmware determines if the cylinder delta is less than or equal to seek range of angular region  $N=1$ . The decision in step 44 determines whether command queue [1] is in the shaded area of angular region  $N=1$ . If it is determined that the cylinder delta is within the shaded area, step 46 requires execution of the command in command queue [1]. Upon execution of the command, the current position of the read/write element is now the physical location of the last LBA of the command just executed (i.e. command queue [1]). In step 48, command queue [1] is removed from the command queue array.

If after step 40, the firmware determines that the rotational latency is not within  $N/10$ ths to  $N+1/10$ ths of a revolution away from the current position of the read/write

THIS SHEET HAS BEEN  
REPLACED

element, or if after step 44 it is determined that the cylinder delta is greater than the seek range [N] of that particular angular region, the command queue index is increased by one in step 50.

In step 52, the firmware verifies that it has checked all commands to determine if they meet the parameters of  $N=1$ . The verification is accomplished by finding if the index number is greater than the number of commands. If the index number is less than the number of commands, the firmware repeats steps 38 et. sequence. If the index number is greater than the number of queued commands, in step 54, the firmware targets the next succeeding angular region  $N=N+1$ .

In step 56, the firmware determines if an entire disc revolution's worth of regions have been searched. In this example, it is determined if  $N=10$ . If an entire disc revolution's worth of regions have not been searched, the method requires the firmware to return to step 36 to begin the search procedure again for the angular region  $N=N+1$ .

If  $N=10$ , however, a new protocol is instigated. More particularly, in step 60, the firmware searches the command queue for the command with the nearest starting cylinder with respect to the current cylinder position of the read/write element. Once this command is found, the "index" of this command is set. Upon completion of step 60, the method requires the firmware to execute the command and remove it from the queue in accordance with steps 46 and 48, respectively.

The method is a practical way to significantly reduce the amount of access time associated with short seeks in memory assemblies which employ a rotational storage medium. By searching the command queue to find and execute the command with the shortest access time, the efficiency of data retrieval is enhanced.

Although the present invention has been described with reference to preferred embodiments, workers skilled in the art will recognise that changes may be made in form and detail without departing from the spirit and scope of the invention.

This invention provides in a system including a data retrieval element and a rotating medium, a method for decreasing rotational latency comprises;



building a seek range array corresponding to discrete angular regions on the medium;

searching a command queue array for a command in a selected discrete angular position; and

executing the command found in the selected discrete angular region of the medium.

This invention provides in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek comprises:

dividing the disc into discrete angular regions;

building a seek range array corresponding to the discrete angular regions;

building a command queue array;

searching successive discrete angular regions for a command in the command queue array; and

executing the command found in a discrete angular region.

This invention provides in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek comprises:

building a seek range array corresponding to the discrete angular regions;

searching a command queue array for a command;

computing a rotational latency from the current position of the read/write element to the command in the command queue;

executing the command found in the command queue; and

removing the executed command from the command queue.

The foregoing description has been given by way of example only and it will be appreciated by a person skilled in the art that modifications can be made without departing from the scope of the present invention.

**CLAIMS:**

1. In a system including a data retrieval element (12) and a rotating medium (10), a method for decreasing rotational latency is characterised by comprising ;

building a seek range array corresponding to discrete angular regions (14) on the medium;

searching a command queue array for a command in a selected discrete angular region; and

executing the command found in the selected discrete angular region of the medium.

2. The method of claim 1 wherein building a seek range array corresponding to discrete regions on the medium comprises:

determining a number of cylinders which may be traversed by the data retrieval element during one revolution of the medium, dividing the media into a plurality of discrete angular regions, and assigning cylinders of the number of cylinders to respective angular regions.

3. The method of claim 1 or claim 2 and further comprising executing a command having a shortest access time with respect to a current position of the data retrieval element.

4. In a disc drive system including a read/write element (12) and a rotating disc (10), a method to decrease access time during a short seek is characterised by comprising:

dividing the disc into discrete angular regions (14);

building a seek range array corresponding to the discrete angular regions;

building a command queue array;



searching the command queue array for a command in successive discrete angular regions; and

executing the command found in a discrete angular region.

5. In a disc drive system including a read/write element (12) and a rotating disc (10), a method to decrease access time during a short seek is characterised by comprising:

building a seek range array corresponding to the discrete angular regions (14);

searching a command queue array for a command;

computing a rotational latency from the current position of the read/write element to the command in the command queue; and

executing the command found in a discrete angular region.

6. The method of claim 4 or claim 5 wherein the seek range represents a number of cylinders which may be crossed by the read/write element within a revolution of the disc.

7. The method of any one of claims 4 to 6 and further comprising executing a command having a shortest access time with respect to a current position of the read/write element.

8. The method of any preceding claim wherein executing the command comprises determining if the command is within a seek range.

9. The method of any preceding claim and further comprising removing the executed command from the command queue array.

10. In a system including a rotating medium having a plurality of tracks, a transducing element adjacent the rotating element and a command queue array containing commands identifying locations on tracks, a process of moving the transducing element



from adjacent a location on a first track to adjacent a location on a different track identified by a command with minimal rotational latency characterised by comprising:

building a seek range array defining locations on plural tracks that may be accessed within one revolution of the medium as the transducing element is moved from the first track;

selecting a location in the seek range array identified by a command in the command queue array; and

moving the transducing element to the selected location.

11. The process of claim 10 wherein building the seek range array comprises:  
dividing the medium into a plurality of discrete angular regions, and  
for each angular region, identifying a number of tracks on the medium that may be traversed by the transducing element from the location on the first track during one revolution of the medium.

12. The process of claim 10 and wherein the location in the seek range array is selected as having a shortest access time with respect to the location on the first track.

13. In a disc drive system including a transducing element and a rotating media, a process to decrease access time during a short seek characterised by comprising:  
building a seek range array defining locations on plural tracks that may be accessed within one revolution of the media as the transducing element is moved from the first track;  
identifying commands in a command queue array addressing locations in the seek range array;  
computing rotational latencies from a current location of the transducing element to locations addressed by identified commands in the command queue array; and  
selecting the identified command having the smallest rotational latency.



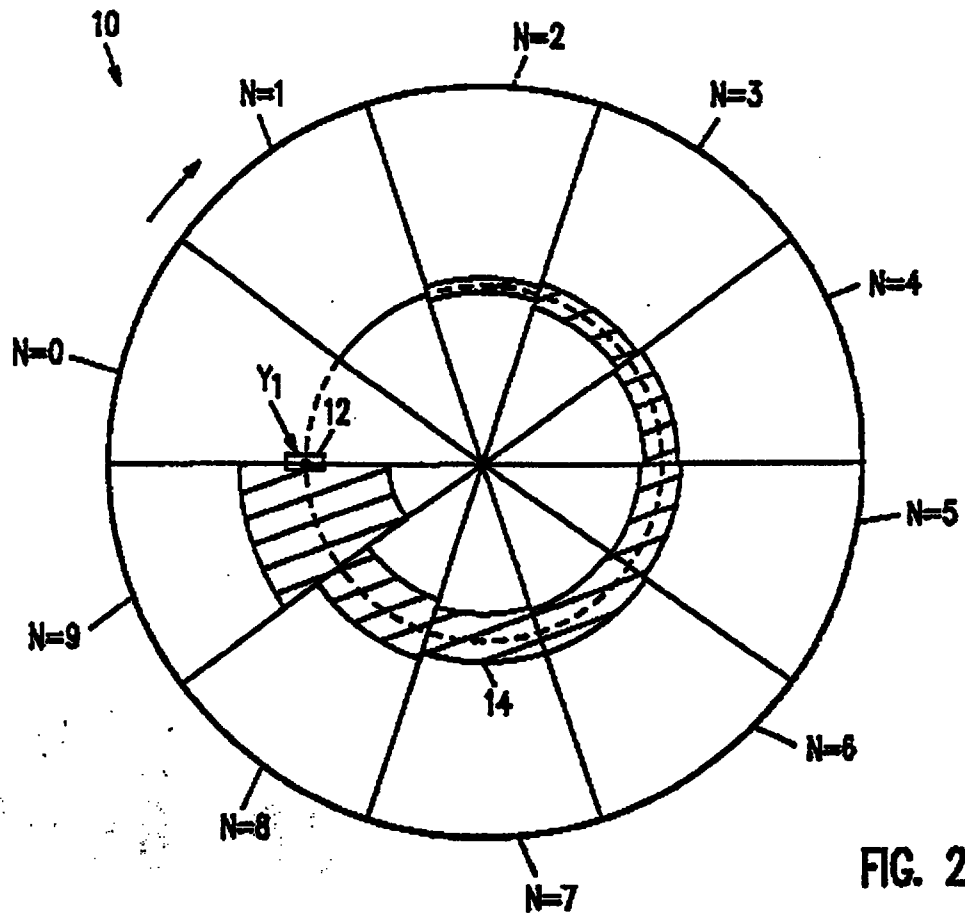
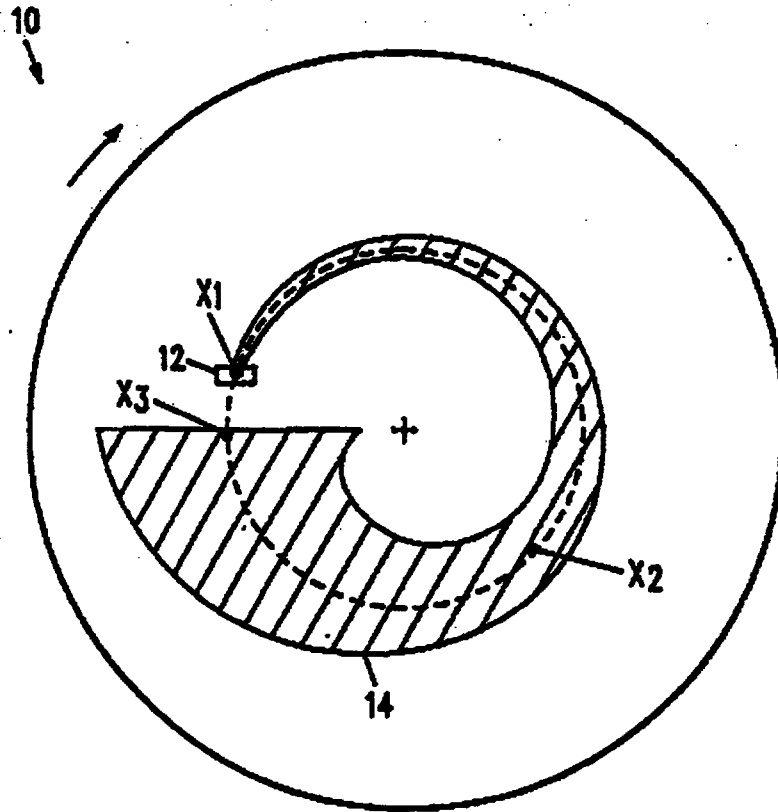
14. The process of claim 13 wherein building the seek range array comprises:  
dividing the media into a plurality of discrete angular regions; and  
for each angular region, identifying a number of tracks on the medium that may be traversed by the transducing element from the current location during one revolution of the medium.
15. The process of claim 13 and further comprising removing the identified command from the command queue array.
16. The method of any one of claims 4 to 7 wherein the seek range array represents the cylinders which may be crossed by the read/write element within the time taken to rotate the disc through the respective discrete angular regions.
17. The method of any preceding claim wherein a command queue array comprises each command in the command queue.

0-327-101

**ABSTRACT****METHOD FOR REDUCING ROTATIONAL LATENCY  
IN A DIC DRIVE**

A method for reducing rotational latency in a system employing a rotating medium (10) includes dividing the medium into discrete angular regions ( $N=0...N=9$ ), building a seek range array corresponding to the discrete angular regions, searching a command queue array for a command, executing the command and removing the executed command from the command queue array. In this manner, commands within the shortest access time of a current position (Y1) of a data retrieval element (12) are executed.

[Fig. 2]



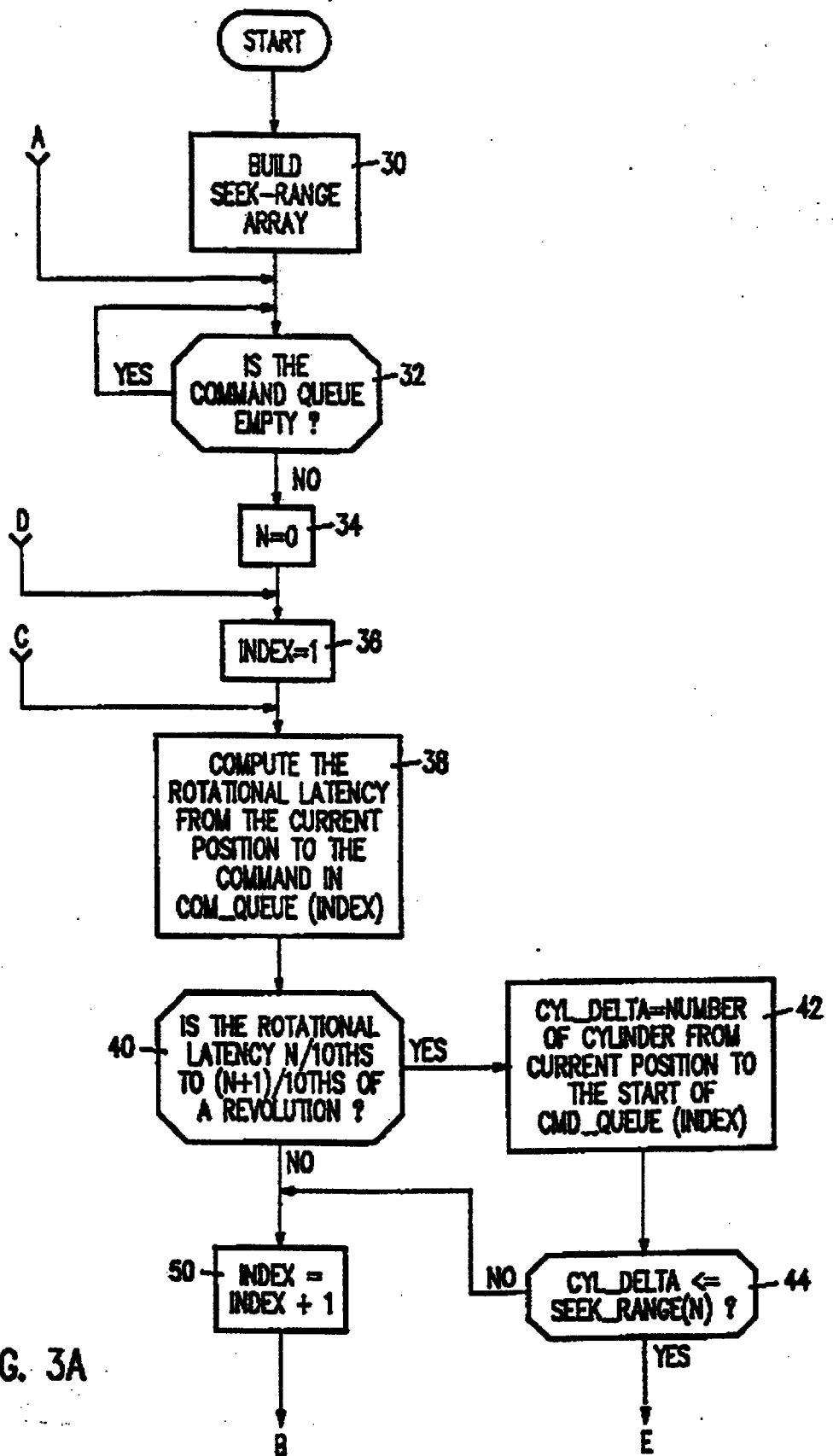


FIG. 3A

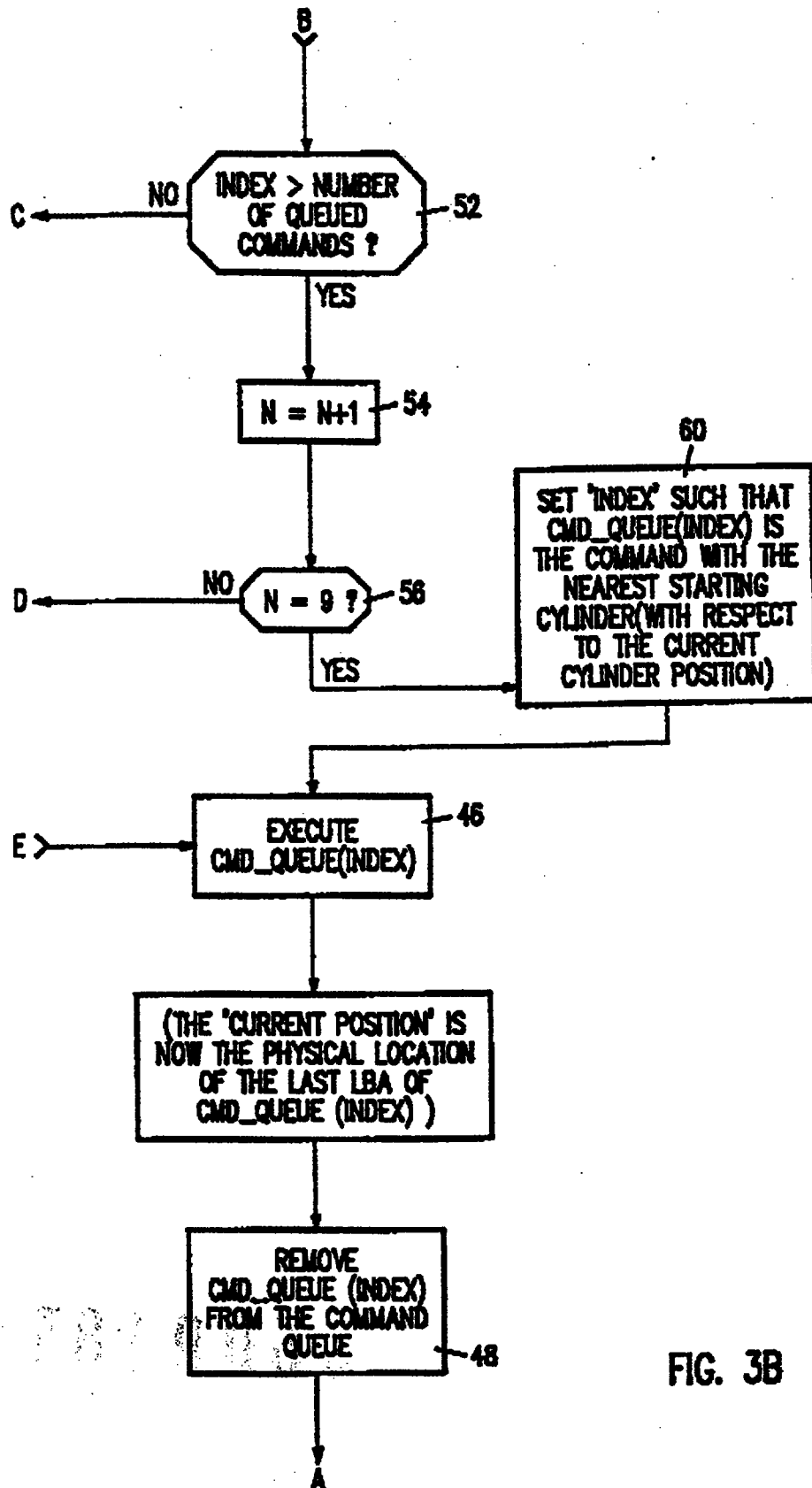


FIG. 3B



29 SEP 1995

1

METHOD FOR REDUCING ROTATIONAL LATENCY  
IN A DISC DRIVE

The present invention relates to seek operations in rotating media and in particular in rotating disc drive memory assemblies. More particularly, the present invention relates to a method for sorting the command queue to reduce rotational latency in disc drive memory assemblies.

Rotating disc memories include one or more discs driven about a vertical spindle axis. Each disc has a plurality of concentric tracks on one or both surfaces from which information is read or onto which information is written by a read/write element. In a disc drive system, a disc rotates at a high speed while the read/write element "flies" over the surface of the rotating disc. The read/write element is positioned over specific areas of the disc in accordance with commands received from the host computer.

The SCSI interface protocol provides a mechanism for host computers to send multiple commands to a disc drive. It allows the disc drive to determine the optimum order of execution for the commands. The set of commands available for execution by the disc drive is referred to as the "command queue".

Traditionally, disc drives have sorted the command queue in an order that minimizes the seek time between the various commands. However, seek time is only one of two components of the true access time of a command. The other component is the rotational latency time or the amount of time the disc drive spends waiting for the appropriate data to rotate under the read/write element. The rotational latency time is a significant part of the total access time for all but the longest seeks. On average, it is the dominant component of the total access time for relatively short seeks.

To determine the optimum order of execution of a set of commands, it is desirable to consider the full access time between commands rather than simply the seek time. An ideal sorting algorithm would sort the command queue in such a way that the sum of the access times of all of the commands in the queue is minimized. This would involve computing the sum of the access times for each of the possible orders of the

This Sheet Has Been Replaced



THIS SHEET HAS BEEN  
REPLACED

REPLACEMENT SHEET

29 SEP 1995

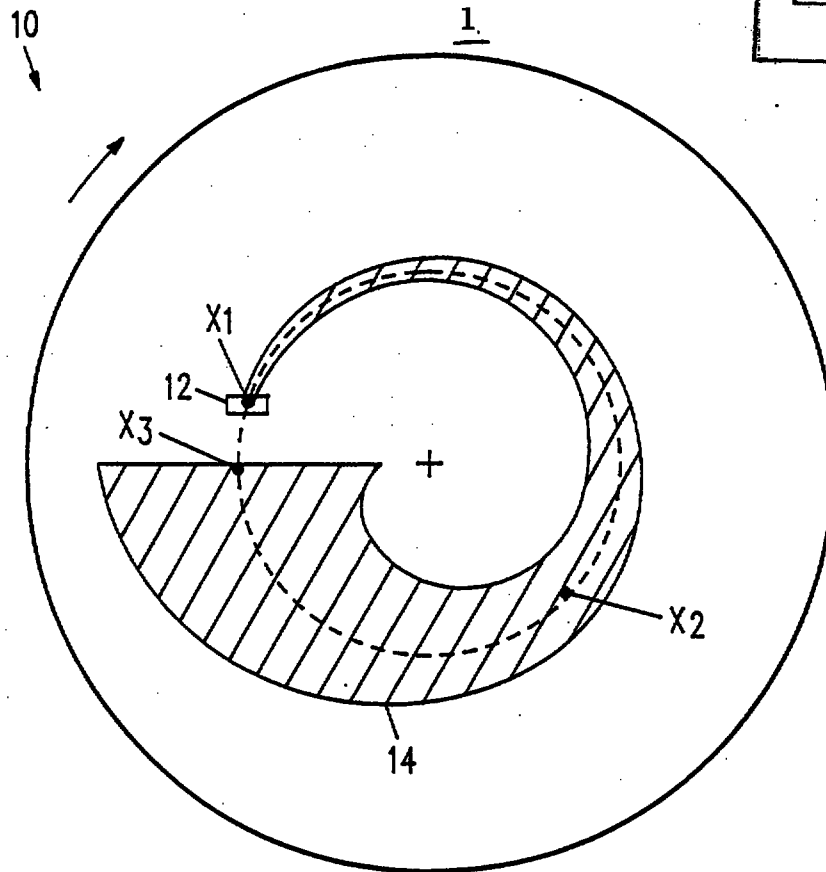


FIG. 1

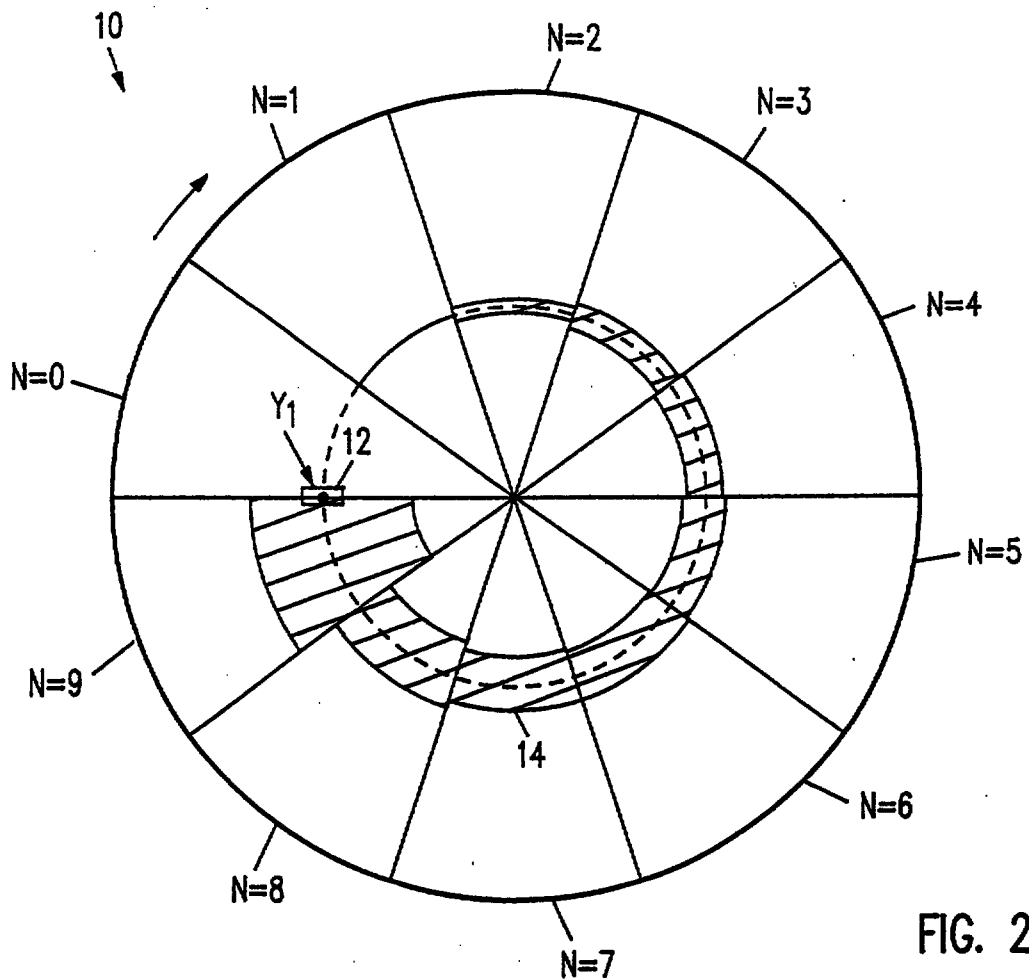


FIG. 2



29 SEP 1995

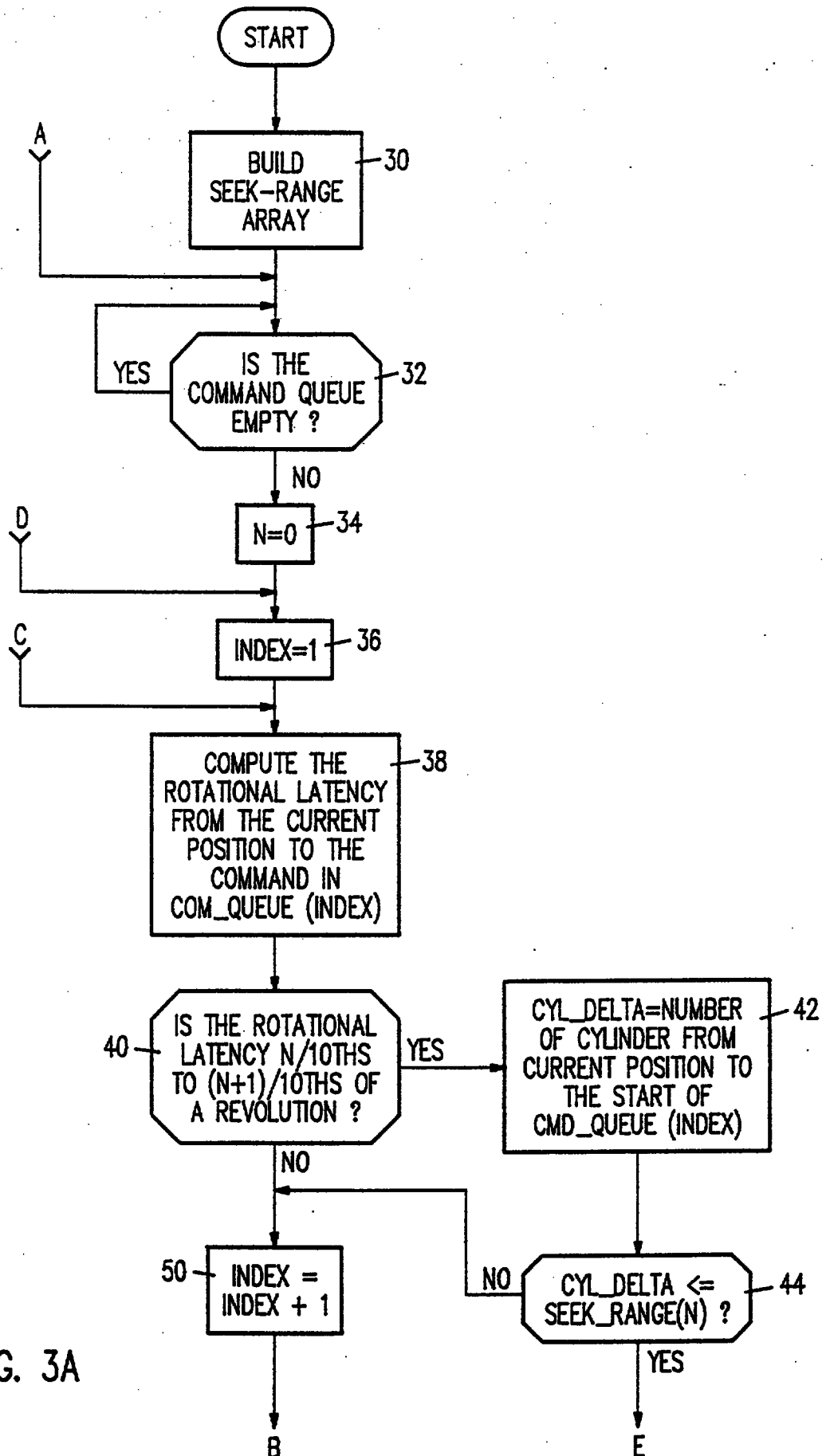
2

FIG. 3A

29 SEP 1995

3.

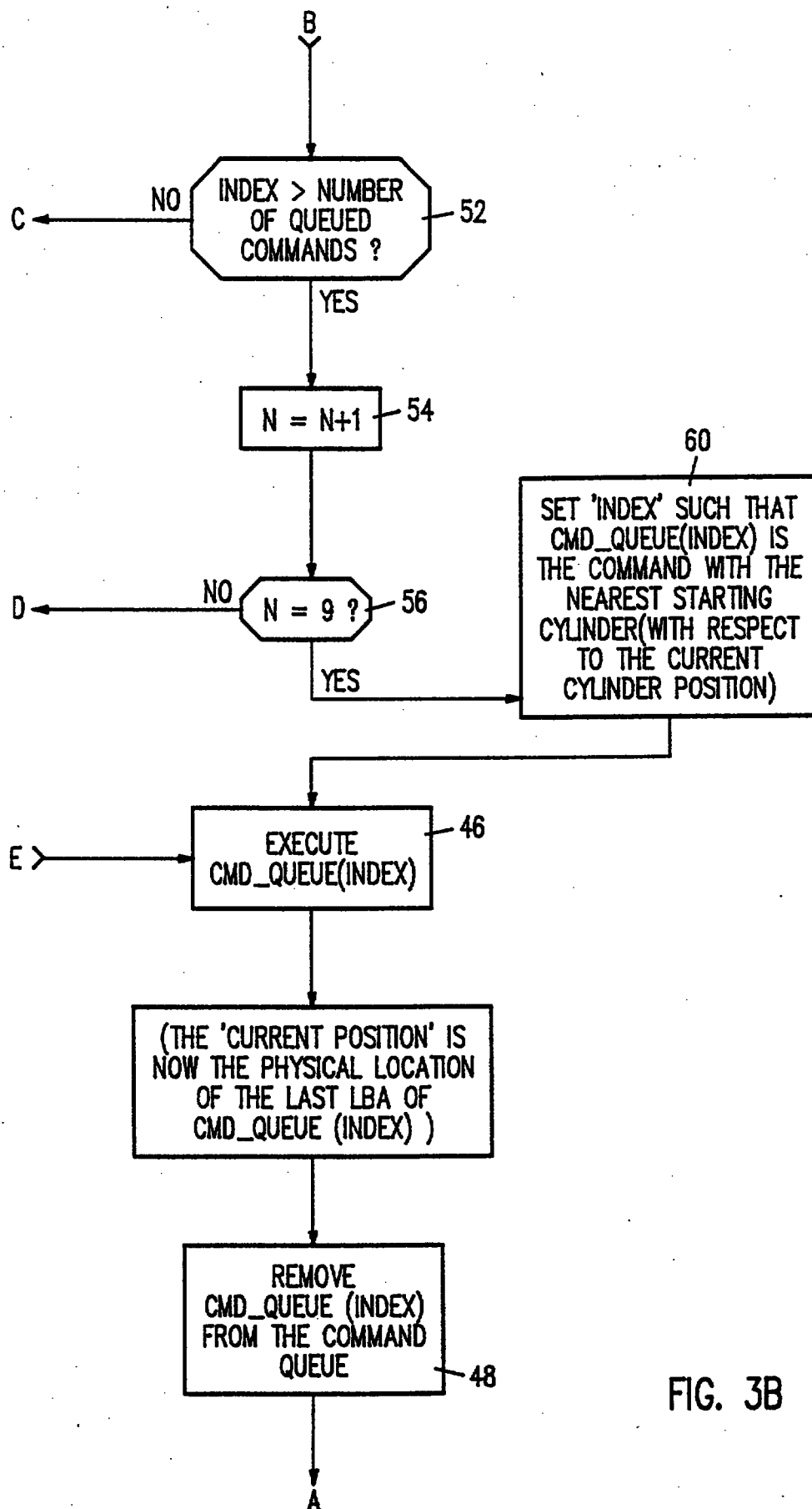


FIG. 3B

This Sheet Has Been Replaced



THIS SHEET HAS BEEN  
REPLACED

REPLACEMENT SHEET

29 SEP 1995

## ABSTRACT

### METHOD FOR REDUCING ROTATIONAL LATENCY

#### IN A DISC DRIVE

A method for reducing rotational latency in a system employing a rotating medium (10) includes dividing the medium into discrete angular regions ( $N=0 \dots N=9$ ), building a seek range array corresponding to the discrete angular regions, searching a command queue array for a command, executing the command and removing the executed command from the command queue array. In this manner, commands within the shortest access time of a current position (Y1) of a data retrieval element (12) are executed.

(Fig. 2)

16 APR 1997

## METHOD FOR REDUCING ROTATIONAL LATENCY IN A DISC DRIVE

The present invention relates to seek operations in rotating media and in particular in rotating disc drive memory assemblies. More particularly, the present invention relates to a method for sorting the command queue to reduce rotational latency in disc drive memory assemblies.

Rotating disc memories include one or more discs driven about a vertical spindle axis. Each disc has a plurality of concentric tracks on one or both surfaces from which information is read or onto which information is written by a read/write element. In a disc drive system, a disc rotates at a high speed while the read/write element "flies" over the surface of the rotating disc. The read/write element is positioned over specific areas of the disc in accordance with commands received from the host computer.

The SCSI interface protocol provides a mechanism for host computers to send multiple commands to a disc drive. It allows the disc drive to determine the optimum order of execution for the commands. The set of commands available for execution by the disc drive is referred to as the "command queue".

Traditionally, disc drives have sorted the command queue in an order that minimizes the seek time between the various commands. However, seek time is only one of two components of the true access time of a command. The other component is the rotational latency time or the amount of time the disc drive spends waiting for the appropriate data to rotate under the read/write element. The rotational latency time is a significant part of the total access time for all but the longest seeks. On average, it is the dominant component of the total access time for relatively short seeks.

To determine the optimum order of execution of a set of commands, it is desirable to consider the full access time between commands rather than simply the seek time. An ideal sorting algorithm would sort the command queue in such a way that the sum of the access times of all of the commands in the queue is minimized. This would

16 APR 1997

involve computing the sum of the access times for each of the possible orders of the queue. Computing access times for every possible order, however, is far too computationally intensive to do in a real-time system such as a disc drive. For example, there are over three million possible orders in a set of ten commands. Modern disc drives can have up to 64 or more commands in the command queue. In addition, the command queue in practical applications is dynamic. In other words, new commands are constantly received from the host computer. Thus, the optimum order of execution of previously received commands often changes, requiring the entire queue to be re-sorted.

Since the above-mentioned ideal algorithm is impractical, alternative methods of sorting have been devised. Several alternative disc scheduling algorithms are discussed in Seltzer et al., "disc scheduling revisited", Usenix Winter 1990 Technical Proceedings, pp. 313-324, January 22, 1990 (hereinafter "Disc Scheduling"). In addition to discussing the first come, first serve algorithms and the above described shorter seek first algorithm, the disc scheduling article discusses SCAN, CSCAN, shortest time first, grouped shortest time first and the weighted shortest time first algorithms. The scan scheduling algorithm orders commands by cylinder number and services all commands for a given cylinder before moving the head to the next cylinder. With the scan algorithm, commands addressing locations on ends of the disc have poor response time because the head must sweep across the disc twice before returning. The CSCAN or cylindrical scan algorithm alleviates this problem by paying one large seek at the end of the disc to move the head all the way to the other end of the disc.

The shortest time first algorithm selects the commands based upon the total access time including seek time and rotational latency. However, as discussed above, the shortest time first algorithm is impractical due to the large computation requirements. The grouped shortest time first algorithm divides the disc into a number of cylinder groups. The shortest time first algorithm is applied to a selected cylinder group to service commands before advancing to the next cylinder group. The weighted shortest time first algorithm applies the standard shortest time first technique but applies an ageing function to the time determined by the computer. For each shortest time first

16 APR 1997

calculation, the total access time is multiplied by a weighting value  $w$ . The value  $w$  is computed by calculating how much time is left before the command will exceed a maximum allowed response time.

Another such disc scheduling algorithm is the shortest access time first algorithm discussed in Jacobson et al., "Disc Scheduling Algorithm Based On Rotational Positions", Hewlett-Packard Laboratories Technical Report, HPL-CSP-91-7, February 26, 1991, pp. 5-18. The shortest access time first algorithm scans the command queue of pending commands and calculates an access time for each command given the known position of the read/write element (if the disc is idle), or the expected position of the read/write element at the end of the current command. The access time is calculated by using a static array containing seek times for each possible seek distance. To reduce the number of commands which must be scanned and for which an access time must be calculated, the shortest access time first algorithm divides the disc into a number of bins. The commands in the command queue are grouped based on the location addressed by each particular command. The bins are then searched in an order determined by a current position of read/write element. The order of bins to be searched based upon the current location of the read/write element is precomputed and stored in an array indexed by the position of the read/write element. The list of bin orderings may require about eight kilobytes of memory for a disc divided into 64 bins.

To select a command for execution, the algorithm calculates an access time for each of the commands grouped into the first non-empty bin found. After the access time for each of the commands is calculated, the command having the shortest access time is generally executed. However, some bins contain cylinders on both sides of the rotational latency line or head trajectory. In other words, some bins may contain only command addresses that can only be accessed in two revolutions. To cope with scenarios where the first non-empty bin contains only command addresses outside of the rotational latency line or head trajectory, the algorithm additionally requires that a threshold value be calculated and stored for each bin or cell. The threshold value represents the largest possible access time a command may have to be executed. After an access time is

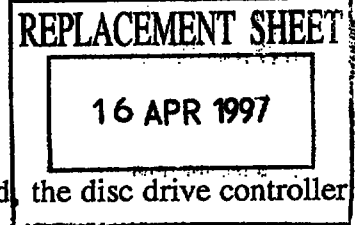
16 APR 1997

calculated for each and every command location with the particular bin being scanned, the best or smallest access time of all the commands is compared with the threshold value. If the best access time of all the commands is below the threshold value, the command having the lowest access time is executed. However, if the best access time of all the command addresses within the bin is greater than the threshold value, the algorithm repeats by scanning all of the commands and calculating an access time for all of the commands in the next non-empty bin.

Although the shortest access time first scheduling algorithm is an improvement over sorting algorithms which merely minimize seek time between commands, the shortest access time first algorithm has several drawbacks. First, the shortest access time first method is extremely time consuming for commands which address clustered locations on the disc. In the operation of many systems, the command queue may, at any one moment, contain a set of commands which address locations that are clustered in a particular area of the disc. As a result, a single bin may contain an extremely large number of command addresses. Because the shortest access time first algorithm requires that the access time be calculated for each and every command address within the particular bin, the algorithm is extremely time consuming for a bin containing clustered addresses for a large number of commands.

Second, the shortest access time first algorithm requires a significant amount of memory. As discussed above, the shortest access time first algorithm requires an array indexed by a position of the read/write element which lists a bin search order for each possible starting bin. In addition, the algorithm further requires an array containing threshold values for each bin. The eight kilobytes of memory required for the array is significant compared to the amount of memory available on some disc drives.

Another improved method of sorting would be to choose the next command to be executed by searching the command queue to find the command with the shortest access time. Although this does not result in a guaranteed minimum sum of all command access times, it comes relatively close.



To compute the access time of an arbitrary command, the disc drive controller firmware must first compute the seek time for the command. The seek time is the time required to move the read/write element from the present track to the desired track. Next, it must determine the expected rotational position of the disc drive after the seek is complete. Since different disc drives will have different seek characteristics, the firmware must provide for some margin of error when computing seek times. For very long seeks, this margin of error may be quite large and hence, impractical for implementation of the present invention. A sorting algorithm which minimizes rotational latency for commands with relatively short seeks, however, would be a significant contribution to the art.

In a first aspect, this invention provides, in a system including a data retrieval element and a rotating medium, a method for decreasing rotational latency characterised by comprising:

- dividing the medium into a plurality of discrete angular regions;
- for each discrete angular region, identifying a number of cylinders that may be traversed by the data retrieval element during a single revolution of the medium;
- assigning a rotational latency to each discrete angular region based on a current location of the data retrieval element;
- searching a command queue array for a command that addresses a location within a range of cylinders determined by the number of cylinders in the discrete angular region having the smallest rotational latency; and
- executing the command addressing a location in the range of cylinders in the selected discrete angular region of the medium.

In a second aspect, this invention provides, in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek characterised by comprising:

- dividing the disc into a plurality of discrete angular regions;
- assigning a rotational latency to each discrete angular region based on a current location of the read/write element;

16 APR 1997

for each discrete angular region, defining a seek range by identifying a number of cylinders that may be traversed by the read/write element during the time to rotate the disc through a predetermined number of discrete angular regions on the disc;

searching a command queue array for a command addressing a location on the disc within a range of cylinders determined by the number of cylinders and in the discrete angular region having the smallest rotational latency; and executing the command found in a discrete angular region.

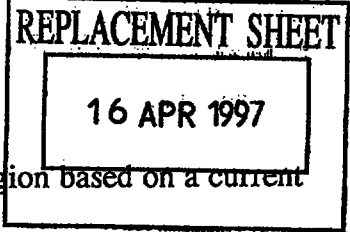
In a third aspect, this invention provides, in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek characterised by comprising:

- (a) dividing the disc into a plurality of discrete angular regions;
- (b) assigning a rotational latency to each discrete angular region based on the current position of the read/write element;
- (c) defining a seek range by identifying, for each discrete angular region, a number of cylinders that may be traversed by the read/write element during the time to rotate the disc through a predetermined number of discrete angular regions on the disc;
- (d) searching a command queue array for a command addressing a location in the seek range and in the discrete angular region having the lowest rotational latency;
- (e) moving the read/write element relative to the disc to the location addressed by the command found in the discrete angular region and executing the command; and
- (f) removing the command from the command queue.

In a fourth aspect, this invention provides, in a disc drive system including a transducing element and a rotating media, a process to decrease access time during a short seek characterised by comprising:

dividing the media into a plurality of discrete angular regions; and

for each angular region, identifying a number of tracks on the medium that may be traversed by the transducing element from the current location during one revolution of the medium;



assigning a rotational latency to each discrete angular region based on a current location of the transducing element;

identifying commands in a command queue array addressing locations in the seek range; and

selecting an identified command addressing a location in the discrete angular region having the smaller rotational latency.

The present invention is a method to minimize rotational latency for short seeks. More particularly, the method requires the firmware to search the command queue for commands which have a starting logical block address (LBA) that is 0.1 to 0.2 of a revolution away from the current rotational position of the read/write element. If one of these commands has a starting cylinder number which is close enough to the current cylinder position of the read/write element to allow a seek to complete within 0.1 of a revolution, then that particular command is executed. Otherwise, the firmware searches the command queue for commands with a starting LBA that is within 0.2 to 0.3 of a revolution away from the current position of the read/write element. If a command can be completed within a seek time of 0.2 of a revolution or less, then that particular command will be executed. If no such command is available, the firmware searches for commands with a starting LBA that is 0.3 to 0.4 of a revolution away, etc. If no commands are found which can be accessed within one revolution of the disc, the firmware searches the command queue for the command which has the shortest expected seek time.

The firmware must have some method of determining the difference in rotational position between two LBAs. Although this could be accomplished by converting the LBA numbers into a physical address (i.e. cylinder, head, sector) the sector numbers of LBAs in different recording zones cannot be compared easily. To resolve this situation, the method requires the firmware to divide the disc into a number (e.g. 100) of discrete angular regions. For the purpose of sorting the queue, the starting and ending LBA for each command is converted into a cylinder, head, and angular region address. The angular region numbers (e.g.  $N=1 \dots N=100$ ) provide a common system of measurement

for all LBAs regardless of recording zone. These angular region numbers are always treated "modulo 100". For example, the difference in rotational position from angular region 90 to angular region 5 is 15. The firmware treats an angular region difference of 10 to 19 as 0.1 of a revolution and 20 to 29 as 0.2 of a revolution, etc.

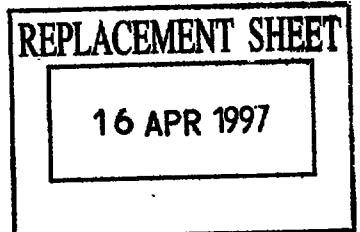
Embodiments of the invention will now be described by way of example only, with reference to the accompanying diagrammatic figures in which;

Figure 1 shows a plan view of a rotating disc, portions of the disc which can be reached within one revolution of the current position of the read/write element are shaded;

Figure 2 shows the rotating disc of Figure 1 divided into discrete angular regions as required by the method of the present invention; and

Figure 3 is a flow diagram of the steps of the method of the present invention and is split between Figures 3A and 3B, identical parts having the same reference numerals throughout.

Figure 1 shows a plan view of rotating disc 10. Read/write element 12 is shown positioned at location  $X_1$  on disc 10. Shaded area 14 represents the seek range [N] or the number of cylinders that can be traversed by read/write element 12 during one disc revolution from its current position at point  $X_1$ . For example, assume that read/write element 12 is able to cross a total of 200 cylinders in one revolution of disc 10. More particularly, within one revolution, starting from point  $X_1$ , read/write element 12 will be able to traverse 200 cylinders. At point  $X_1$ , shaded region 14 is very small because a very small percentage of the disc revolution is complete. At point  $X_2$ , however, half of a revolution has passed. The number of cylinders that read/write element 12 can traverse within half a disc revolution is approximately 80-120 cylinders. Correspondingly, the width of shaded area 14 at point  $X_2$  has significantly increased with respect to the amount of shaded area at point  $X_1$ . At point  $X_3$ , an entire disc revolution is almost complete. Accordingly, the portion of shaded area 14 immediately adjacent point  $X_3$  represents the greatest area of the disc that may be crossed in one revolution (approximately 200 cylinders).



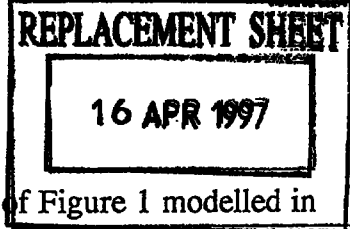


Figure 2 shows rotating disc 10 and shaded area 14 of Figure 1 modelled in discrete angular regions as required by the method of the present invention. More particularly, disc 10 is divided into ten angular regions labelled  $N=1$ ,  $N=2$ ,  $N=3 \dots N=9$ ,  $N=10$  wherein "N" is relative to the current actuator position.

In operation, read/write element 12 may be positioned at point  $Y_1$  which is the boundary between regions  $N=9$  and  $N=10$ . The method requires the firmware to initially search the command queue for an executable command within the angular region labelled  $N=1$ . If a command is found within region  $N=1$ , the firmware then determines if the number of cylinders read/write element 12 must traverse to reach the start of the found command is within the seek range for that particular angular region. If the command is within the seek range (i.e. the shaded area), the command is executed. If no command is found within the seek range, the firmware searches the command queue for a command within the seek range of the next angular region  $N=2$ . Each angular region is searched in a clockwise fashion until an entire revolution of the disc is complete. If no commands can be found within any of the seek ranges (i.e. shaded areas) of each region, the firmware executes the command which requires the shortest seek time with respect to the current position of the read/write element.

Figure 3 is a flow diagram of the steps of the method of the present invention. In step 30, a seek range array is built. The seek range array is indexed according to time in units of tenths of a revolution. Each element of the array contains the number of cylinders which can be crossed in the amount of time corresponding to the element's index. For example, seek range  $[3]=100$  means that a seek can cross 100 cylinders in  $3/10$  of a revolution.

In step 32, the firmware determines if the command queue is empty. As long as the queue is empty, no action will be taken. Note, however, that commands may be added to the command queue at any time. In step 34, it has been determined that the command queue is not empty, so the firmware targets angular region  $N=1$  by initialising  $N$  at 1. In step 36, the firmware accesses the first command (i.e. index=1) of the command queue index. The command queue index is an array that contains each

16 APR 1997

command in the command queue. Thus, command queue [1] is the first command in the queue, command queue [2] is the second command, etc.

In step 38, the firmware computes the rotational latency from the current position of the read/write element to the position of the command in command queue [1].

In step 40, the firmware determines if the rotational latency computed in step 38 is  $N/10$ ths to  $(N+1)/10$ ths of a revolution away from the current position of the read/write element (i.e. within the parameters of region  $N=1$ ). If the rotational latency falls within the parameters which define angular region  $N=1$ , in step 42, the "cylinder delta" is computed. The cylinder delta is the number of cylinders from the current position of the read/write element to the start cylinder of the command in command queue [1].

In step 44, the firmware determines if the cylinder delta is less than or equal to seek range of angular region  $N=1$ . The decision in step 44 determines whether command queue [1] is in the shaded area of angular region  $N=1$ . If it is determined that the cylinder delta is within the shaded area, step 46 requires execution of the command in command queue [1]. Upon execution of the command, the current position of the read/write element is now the physical location of the last LBA of the command just executed (i.e. command queue [1]). In step 48, command queue [1] is removed from the command queue array.

If after step 40, the firmware determines that the rotational latency is not within  $N/10$ ths to  $(N+1)/10$ ths of a revolution away from the current position of the read/write element, or if after step 44 it is determined that the cylinder delta is greater than the seek range [N] of that particular angular region, the command queue index is increased by one in step 50.

In step 52, the firmware verifies that it has checked all commands to determine if they meet the parameters of  $N=1$ . The verification is accomplished by finding if the index number is greater than the number of commands. If the index number is less than the number of commands, the firmware returns to step 38. If the index number is

greater than the number of queued commands, in step 54, the firmware targets the next succeeding angular region  $N=N+1$ .

In step 56, the firmware determines if an entire disc revolution's worth of regions have been searched. In this example, an entire disc revolution's worth of regions has been searched if  $N=11$ . If an entire disc revolution's worth of regions have not been searched (i.e.  $N < 11$ ), the method requires the firmware to return to step 36 to begin the search procedure again for the angular region  $N=N+1$ .

If  $N=11$ , however, a new protocol is instigated. More particularly, in step 60, the firmware searches the command queue for the command with the nearest starting cylinder with respect to the current cylinder position of the read/write element. Once this command is found, the "index" of this command is set. Upon completion of step 60, the method requires the firmware to execute the command and remove it from the queue in accordance with steps 46 and 48, respectively.

The method is a practical way to significantly reduce the amount of access time associated with short seeks in memory assemblies which employ a rotational storage medium. By searching the command queue to find and execute the command with the shortest access time, the efficiency of data retrieval is enhanced.

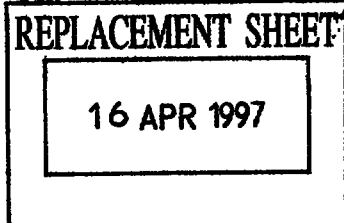
Although the present invention has been described with reference to preferred embodiments, workers skilled in the art will recognise that changes may be made in form and detail without departing from the spirit and scope of the invention.

This invention provides in a system including a data retrieval element and a rotating medium, a method for decreasing rotational latency comprises;

building a seek range array corresponding to discrete angular regions on the medium;

searching a command queue array for a command in a selected discrete angular region; and

executing the command found in the selected discrete angular region of the medium.



This invention provides in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek comprises:

dividing the disc into discrete angular regions;

building a seek range array corresponding to the discrete angular regions;

building a command queue array;

searching successive discrete angular regions for a command in the command queue array; and

executing the command found in a discrete angular region.

This invention provides in a disc drive system including a read/write element and a rotating disc, a method to decrease access time during a short seek comprises:

building a seek range array corresponding to the discrete angular regions;

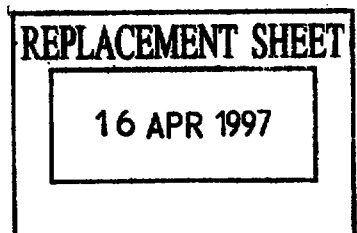
searching a command queue array for a command;

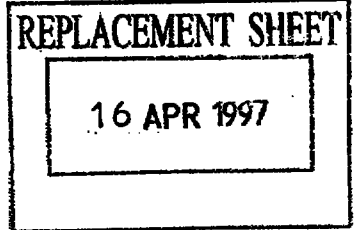
computing a rotational latency from the current position of the read/write element to the command in the command queue;

executing the command found in the command queue; and

removing the executed command from the command queue.

The foregoing description has been given by way of example only and it will be appreciated by a person skilled in the art that modifications can be made without departing from the scope of the present invention.





CLAIMS:

1. In a system including a data retrieval element (12) and a rotating medium (10), a method for decreasing rotational latency is characterised by comprising:
  - dividing the medium into a plurality of discrete angular regions;
  - for each discrete angular region, identifying a number of cylinders that may be traversed by the data retrieval element during a single revolution of the medium;
  - assigning a rotational latency to each discrete angular region based on a current location of the data retrieval element;
  - searching a command queue array for a command that addresses a location within a range of cylinders determined by the number of cylinders in the discrete angular region having the smallest rotational latency; and
  - executing the command addressing a location in the range of cylinders in the selected discrete angular region of the medium.
2. In a disc drive system including a read/write element (12) and a rotating disc (10), a method to decrease access time during a short seek is characterised by comprising:
  - dividing the disc into a plurality of discrete angular regions (14);
  - assigning a rotational latency to each discrete angular region based on a current location of the read/write element;
  - for each discrete angular region, defining a seek range by identifying a number of cylinders that may be traversed by the read/write element during the time to rotate the disc through a predetermined number of discrete angular regions on the disc;
  - searching a command queue array for a command addressing a location on the disc within a range of cylinders determined by the number of cylinders and in the discrete angular region having the smallest rotational latency; and executing the command found in a discrete angular region.



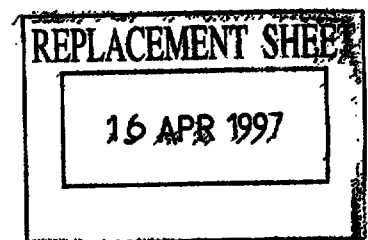
16 APR 1997

3. In a disc drive system including a read/write element (12) and a rotating disc (10), a method to decrease access time during a short seek is characterised by comprising:
- (a) dividing the disc into a plurality of discrete angular regions;
  - (b) assigning a rotational latency to each discrete angular region based on the current position of the read/write element;
  - (c) defining a seek range by identifying, for each discrete angular region, a number of cylinders that may be traversed by the read/write element during the time to rotate the disc through a predetermined number of discrete angular regions on the disc;
  - (d) searching a command queue array for a command addressing a location in the seek range and in the discrete angular region having the lowest rotational latency;
  - (e) moving the read/write element relative to the disc to the location addressed by the command found in the discrete angular region and executing the command; and
  - (f) removing the command from the command queue.
4. The method of claim 2 or claim 3 wherein the seek range represents a number of cylinders which may be crossed by the read/write element within a single revolution of the disc.
5. In a disc drive system including a transducing element (12) and a rotating media (10), a process to decrease access time during a short seek characterised by comprising:
- dividing the media into a plurality of discrete angular regions; and
  - for each angular region, identifying a number of tracks on the medium that may be traversed by the transducing element from the current location during one revolution of the medium;
  - assigning a rotational latency to each discrete angular region based on a current location of the transducing element;
  - identifying commands in a command queue array addressing locations in the seek range; and



selecting an identified command addressing a location in the discrete angular region having the smaller rotational latency.

6. The method of claim 1 wherein identifying the number of cylinders includes:  
creating a seek-range array having stored seek ranges corresponding to each discrete angular region.
7. The method of claim 3 and further including assigning a new rotational latency to each discrete angular region based on the position of the read/write element following movement to the location addressed by the first-named command, repeating step (c) to define a new seek range, searching the command queue array for a second command addressing a location in the new seek range and in the discrete angular region having the new lowest rotational latency, moving the read/write element relative to the disc to the location addressed by the second command, and executing the second command.
8. The process of claim 5 where the identification of commands and selection of identified commands is performed for each angular region in succession.



16 APR 1997

## ABSTRACT

METHOD FOR REDUCING ROTATIONAL LATENCY  
IN A DIC DRIVE

A method for reducing rotational latency in a system employing a rotating medium (10) includes dividing the medium into discrete angular regions ( $N=0 \dots N=9$ ), building a seek range array corresponding to the discrete angular regions, searching a command queue array for a command, executing the command and removing the executed command from the command queue array. In this manner, commands within the shortest access time of a current position (Y1) of a data retrieval element (12) are executed.

[Fig. 2]

1/3

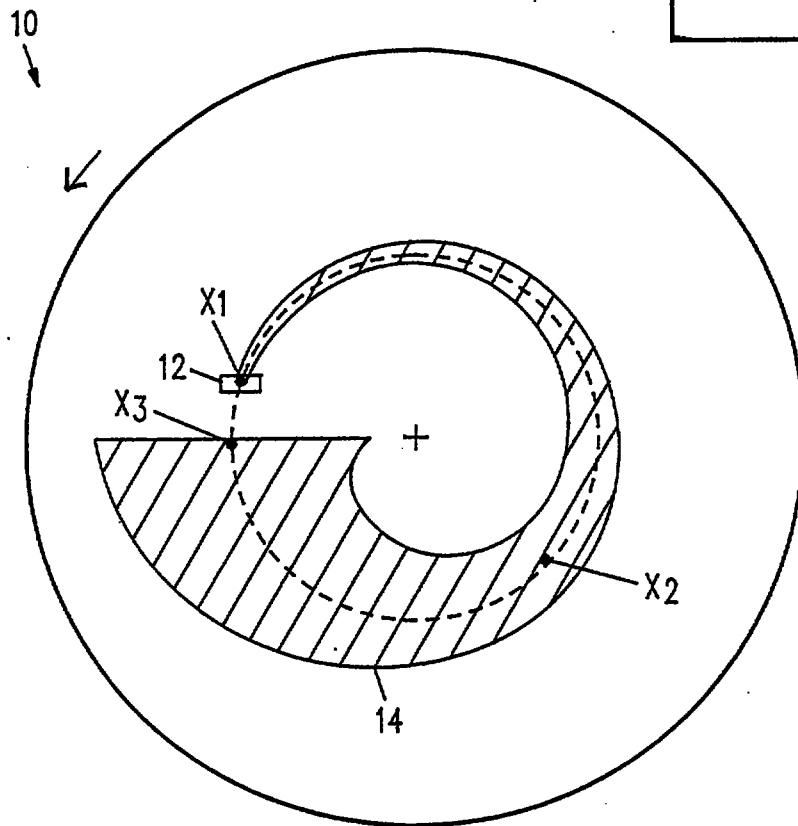


FIG. 1

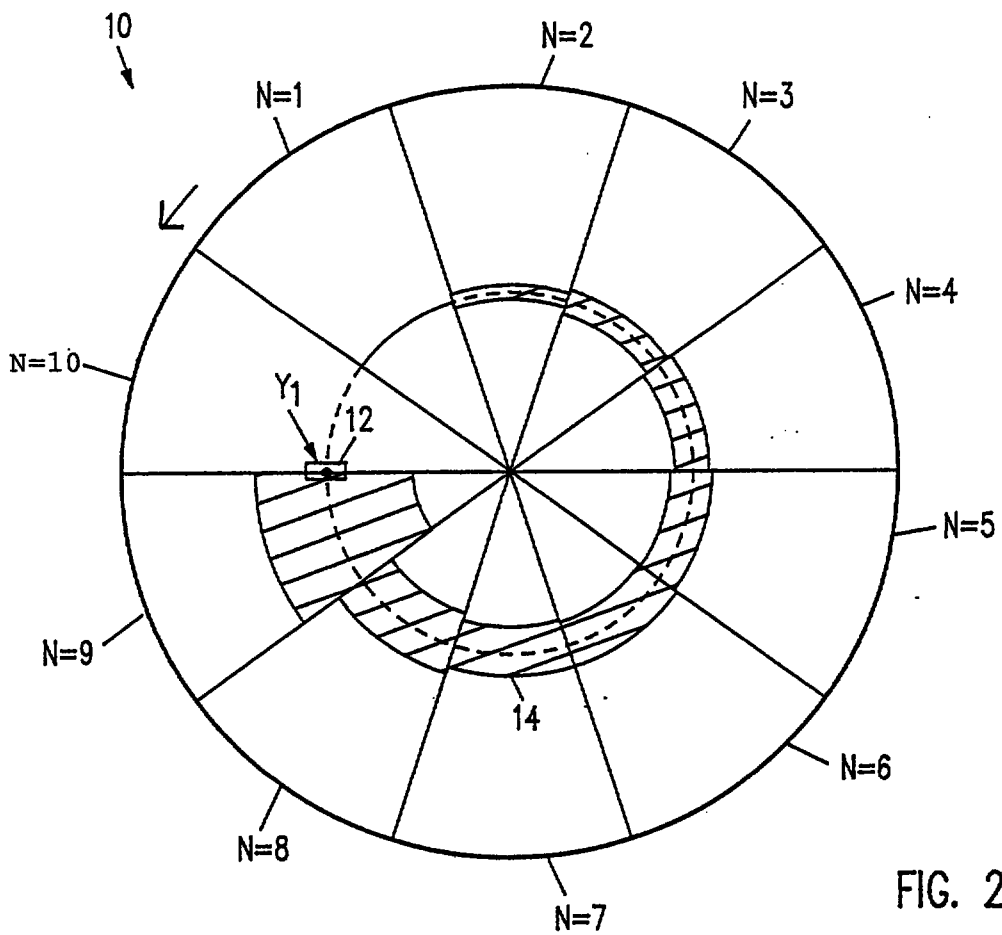
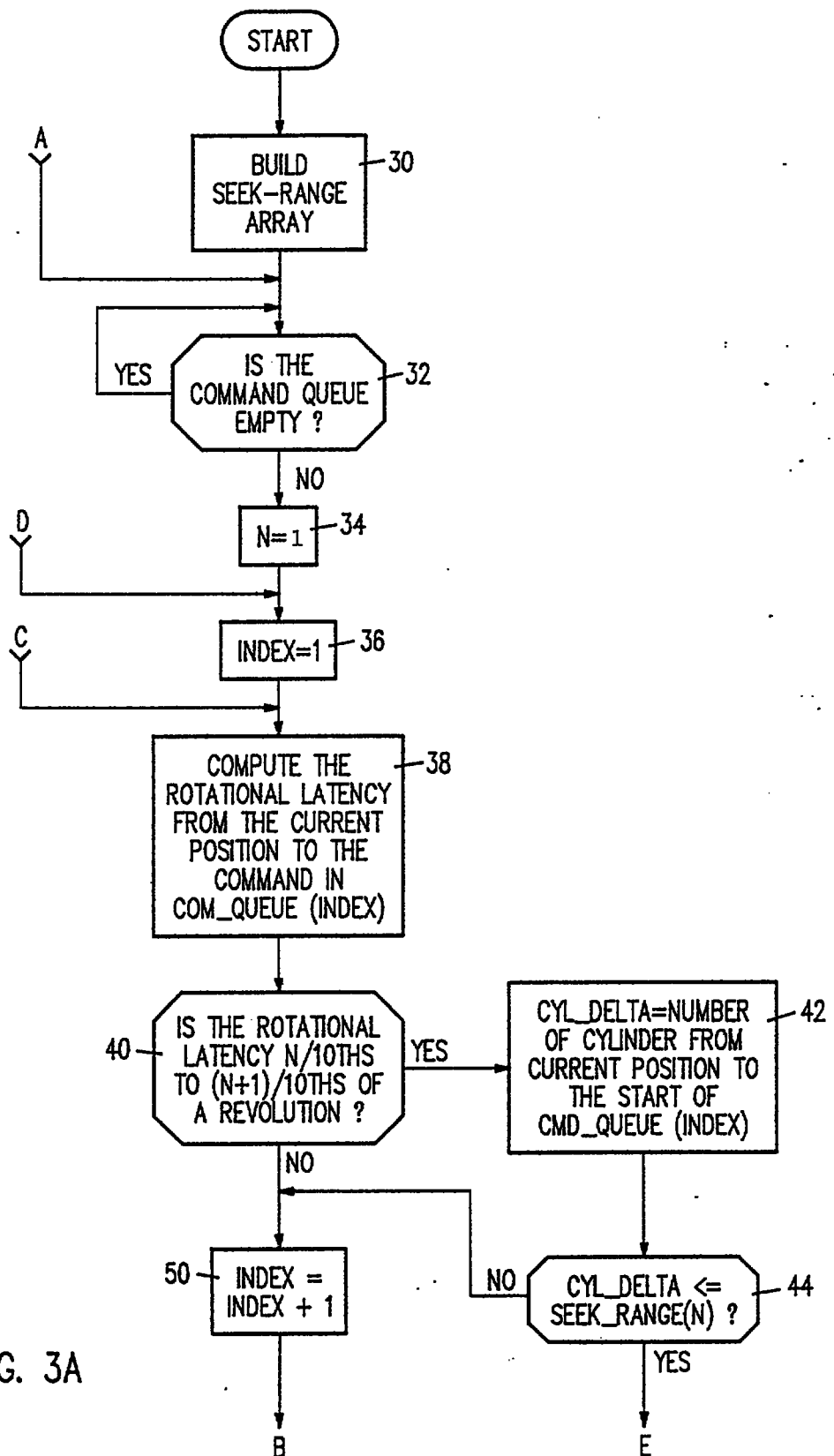


FIG. 2

16 APR 1997

 $\frac{2}{3}$ 

16 APR 1997

