



[12] 发明专利申请公开说明书

[21]申请号 94112983.7

[51]Int.Cl⁶

G11B 20/10

[43]公开日 1996年2月28日

[22]申请日 94.12.19

[30]优先权

[32]93.12.23 [33]US [31]173,541

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 S·R·黑兹勒

W·J·卡贝拉克

[74]专利代理机构 中国专利代理(香港)有限公司

代理人 董巍 马铁良

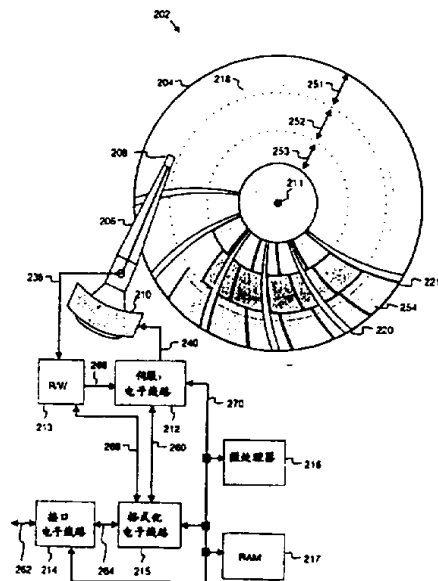
G11B 5/00

权利要求书 9 页 说明书 39 页 附图页数 13 页

[54]发明名称 用于固定块磁盘驱动器的扇区结构

[57]摘要

一种扇区伺服式区位记录磁盘驱动器，它包括扇区结构，在这种结构中，记录头能在不使用数据标识字段但使用来自电子存储器及伺服扇区的信息的情况确定数据扇区的位置并标识该扇区，而所述的伺服扇区则不一定要与前述数据扇区相邻。磁道包含有伺服信息和数据，但不包含有数据扇区标识信息。设置在伺服扇区中的信息包括一伺服扇区号和其它标识，这些标识能使得磁盘驱动器确定记录头的位置并在磁头落在磁道上时不必等待就进行读写操作。



权利要求书

1. 一种供固定块结构磁盘驱动器使用的数据记录磁盘, 磁盘驱动器带有一个能读取位置信息的磁头, 磁盘则被划分成若干径向间隔的磁道, 该磁道中的至少一个被划分成若干有角度的扇区, 这些扇区只包括用于记录用户数据的数据扇区以及伺服扇区, 所述的伺服扇区具有事先记录下来的磁头定位信息以便标识出磁道与伺服扇区的位置, 其中, 所述磁道上伺服扇区的数量不等于该磁道上数据扇区的数量, 而且, 在所述伺服扇区内编码有确定数据扇区圆周位置与特征的信息。

2. 如权利要求 1 所述之数据记录磁盘, 其特征在于, 所确定的数据扇区特征包括数据扇区号。

3. 如权利要求 1 所述的之数据记录磁盘, 其特征在于, 所确定的数据扇区位置包括距前一个伺服扇区的距离。

4. 如权利要求 1 所述之数据记录磁盘, 其特征在于, 编入所述伺服扇区内的信息包括伺服扇区号信息。

5. 如权利要求 4 所述之数据记录磁盘, 其特征在于, 所述伺服扇区号信息包含整个伺服扇区号的一个子集。

6. 如权利要求 1 所述之数据记录磁盘, 其特征在于, 编入所述伺服扇区内的信息包括地址标记。

7. 如权利要求 6 所述之数据记录磁盘, 其特征在于, 在标识所述磁道上固定位置的索引标记中, 编有确定数据扇区之位置与特征的附加信息。

8. 如权利要求 1 所述的之数据记录磁盘, 其特征在于, 至少有某些伺服扇区将至少某些数据扇区划分成主数据段和从数据段, 该主数据段位于上述划分伺服扇区之前并具有自己的圆周位置, 该位置是由按圆周在前的伺服扇区所决定的, 而从数据段则位于上述划分伺服扇区之后并具有自己的圆周位置, 该位置是由上述划分伺服扇区所决定的。

9. 如权利要求 1 所述之数据记录磁盘, 其特征在于, 至少有某些数据扇区沿圆周方向紧随其它数据扇区并具有自己的圆周位置, 该位置是由按圆周在前的伺服扇区所决定的。

10. 包含有固定块结构的伺服磁盘驱动器, 其特征在于, 它包括:

一数据记录磁盘, 此磁盘具有若干径向间隔的磁道, 至少有一个磁道被分成若干有角度的扇区, 这些扇区只包括伺服扇区和数据扇区, 其中, 所述磁道上伺服扇区的数量不等于该磁道上数据扇区的数量;

一记录头, 此记录头用于读取上述伺服扇区中的信息并用于读写上述数据扇区中的用户数据;

用于响应从上述伺服扇区中所读取的信息以便根据数据扇区距所述伺服扇区的距离来确定该数据扇区的位置的装置; 以及

用于响应从上述伺服扇区中所读取的信息以便确定所述数据扇区特征的装置。

11. 如权利要求 10 所述之磁盘驱动器, 其特征在于, 所述的数据扇区是由数据扇区号来标识的。

12. 如权利要求 10 所述之磁盘驱动器, 其特征在于, 所述的磁

道还包括一索引标记,此标记标识出该磁道上的一固定位置,而且,用于确定数据扇区位置的装置和用于识别数据扇区的装置均能响应上述索引标记。

13. 如权利要求 10 所述之磁盘驱动器,其特征在于,从上述伺服扇区中读取出的信息包括伺服扇区号信息。

14. 如权利要求 13 所述之磁盘驱动器,其特征在于,所述的伺服扇区号信息包含着整个扇区号的一个子集。

15. 如权利要求 10 所述之磁盘驱动器,其特征在于,所述用于确定数据扇区位置和特征的装置包括:电子存储器,此存储器包含有扇区结构信息;以及伺服电子线路,此电路根据读取自所述伺服扇区的信息来选择所述电子存储器中扇区结构信息的若干部分并计算从上述伺服扇区到数据扇区的距离,该距离为所选定的扇区结构信息的函数,同时,所述电子线路还用于计算上述数据扇区的特征,此特征也是所选定的扇区结构信息的函数。

16. 如权利要求 15 所述之磁盘驱动器,其特征在于,所述扇区结构信息包括存储段结构。

17. 如权利要求 16 所述之磁盘驱动器,其特征在于,所述存储段结构包括数据段长度。

18. 如权利要求 17 所述之磁盘驱动器,其特征在于,所述数据段长度表示为时钟脉冲数。

19. 如权利要求 16 所述之磁盘驱动器,其特征在于,所述存储段结构包括数据扇区标识。

20. 如权利要求 16 所述之磁盘驱动器,其特征在于,所述包含有存储段结构的电子线路是随机存取存储器或只读存储器。

21. 如权利要求 15 所述之磁盘驱动器, 其特征在于, 所述扇区结构信息还包括存储带结构, 该存储带结构能使上述伺服电子线路确定并识别不紧接在伺服扇区之后的数据扇区。

22. 如权利要求 21 所述之磁盘驱动器, 其特征在于, 所述存储带结构包括同步字段长度、填充字段长度、数据扇区长度、每条磁道上伺服扇区的数量以及每条磁道上数据扇区的数量。

23. 如权利要求 22 所述之磁盘驱动器, 其特征在于, 所述同步字段长度、填充字段长度以及数据扇区长度均体现为时钟脉冲数。

24. 如权利要求 22 所述之磁盘驱动器, 其特征在于, 所述包含有上述存储带结构的电子存储器是专用存储寄存器。

25. 如权利要求 15 所述之磁盘驱动器, 其特征在于, 所述用于确定数据扇区位置和特征的装置还用伺服扇区号和所述扇区结构信息来计算数据扇区号。

26. 如权利要求 10 所述之磁盘驱动器, 其特征在于, 调整所确定的数据扇区特征以补偿柱面与磁头的错位。

27. 在包含有固定块结构的伺服磁盘驱动器中, 用于确定并识别所选定数据扇区的方法, 其特征在于所包括的步骤有: 从在前的伺服扇区中读取信息; 从电子存储器中接收扇区结构信息; 根据所述扇区结构计算出数据扇区号以及计算出从在前的伺服扇区到所述数据扇区起点的距离, 此距离是前述扇区结构信息的函数; 而所说的伺服磁盘驱动器带有: 一数据记录磁盘, 此磁盘具有径向间隔的磁道和按圆周间隔的有角度的扇区, 这些扇区包括数据扇区和伺服扇区; 以及一记录头, 此记录头用于读取上述伺服扇区内的信息并读写上述数据扇区内的用户信息。

28. 如权利要求 27 所述之方法, 其特征在于, 当每个数据扇区到达上述记录头时, 均连续地计算数据扇区的位置和特征, 直至确定并识别出了所选择的数据扇区。

29. 如权利要求 27 所述之方法, 其特征在于, 在所选定的数据扇区到达上述记录头之前计算出所选定的数据扇区的位置和特征, 其中, 用一扇区脉冲来给出所述数据扇区到达了记录头的信号。

30. 如权利要求 27 所述之方法, 其特征在于, 所计算出的距离体现为时钟脉冲数。

31. 如权利要求 27 所述之方法, 其特征在于, 它在计算出所述的数据扇区号之后还包括调整该数据扇区号以补偿柱面及磁头错位这一步骤。

32. 如权利要求 31 所述之方法, 其特征在于, 所说的调整步骤包括用一错位值来修正所说的数据扇区号。

33. 包含有固定块结构的伺服磁盘驱动器, 其特征在于它包括:

一数据记录磁盘, 此磁盘带有径向间隔的磁道以及按圆周间隔的有角度的扇区, 这些扇区包括数据扇区和伺服扇区;

一记录头, 此记录头用于读取所述伺服扇区内的信息并读写上述数据扇区内的用户数据;

一伺服电子线路, 该线路用于根据读取自单个伺服扇区内的信息来确定至少两个后续数据扇区的位置和特征, 而且, 此线路还用于根据读取自同一伺服扇区内的信息来确定该伺服扇区是否将一数据扇区分割开来了。

34. 如权利要求 33 所述之磁盘驱动器, 其特征在于, 用上述读取自该单个伺服扇区内的信息去从电子存储器中检索附加信息。

35. 供固定块结构磁盘驱动器使用的数据记录磁盘中的磁道结构, 所述磁盘驱动器带有一个能读取定位信息的磁头, 而所述磁盘则具有径向间隔的磁道和按圆周间隔的有角度的扇区, 所说的磁道结构将一条磁道分成若干个重复的存储段, 每个存储段均具有多个等距间隔的伺服扇区和多个数据扇区, 至少有一个数据扇区被伺服扇区所分开, 上述数据扇区不包含有用于确定该数据扇区圆周位置的数据标识信息以及用于标识该数据扇区之数据扇区号的数据标识信息。

36. 如权利要求 35 所述之数据记录磁盘, 其特征在于, 包括有定位信息的伺服扇区的后面是: 伺服填充字段; 包括数据填充字段的数据扇区以及至少一个包括分割填充字段的被分割开来的数据扇区。

37. 供固定块结构磁盘驱动器使用的数据记录磁盘中的磁道结构, 所述磁盘驱动器带有一个能读取定位信息的磁头, 而所述磁盘则具有径向间隔的磁道和按圆周间隔的有角度的扇区, 所说的磁道结构将一条磁道分成若干个重复的存储段, 每个存储段均具有多个等距间隔的伺服扇区和多个数据扇区, 至少有一个数据扇区被伺服扇区所分开, 上述数据扇区不包含有用于确定该数据扇区圆周位置的信息以及用于标识该数据扇区之数据扇区号的信息, 而所述伺服扇区则包含有用于确定该数据扇区圆周位置的信息和用于标识该数据扇区之数据扇区号的信息。

38. 如权利要求 37 所述之数据记录磁盘, 其特征在于, 所述包括定位信息的伺服扇区的后面是: 伺服填充字段; 包括数据填充字段的数据扇区以及至少一个包括分割填充字段的被分割开来的数据扇

区。

39. 如权利要求 35 所述之数据记录磁盘, 其特征在于, 至少有一个存储段包括有一存根字段。

40. 包含有固定块结构的伺服磁盘驱动器, 其特征在于它包括:

一数据记录磁盘, 此磁盘具有径向间隔的磁道和按圆周间隔的有角度的扇区, 这些扇区包括数据扇区和伺服扇区, 其中, 对至少一条磁道来说, 数据扇区的数量不等于伺服扇区的数量;

一记录头, 此记录头用于读取上述伺服扇区内的定位信息和读写上述数据扇区内的用户数据;

一磁头定位系统, 其中, 利用格式化电子线路和伺服电子线路相配合来识别上述数据扇区并确定该扇区的位置, 在这里, 所述格式化电子线路将一逻辑块地址转换成数据扇区号, 而所述伺服电子线路则利用该数据扇区号提供一字节时钟脉冲数, 此脉冲数表示从在前的伺服扇区到由上述数据扇区号所标识的数据扇区的距离。

41. 如权利要求 40 所述之磁盘驱动器, 其特征在于, 所述伺服电子线路包括扇区脉冲逻辑电路, 此电路可以存取一格式表, 该格式表包含有用于每个轴芯(core)的存储段信息。

42. 如权利要求 41 所述之磁盘驱动器, 其特征在于, 所述扇区脉冲逻辑电路包括: 一数据计数器; 一填充计数器以及一同步计数器, 此同步计数器对字节时钟脉冲进行计数, 而该字节时钟脉冲则用于确定数据扇区的位置并标识该扇区。

43. 如权利要求 42 所述之磁盘驱动器, 其特征在于, 所述扇区脉冲逻辑电路利用接收自上述伺服扇区计数器的伺服扇区号来访问所说的格式表。

44. 如权利要求 41 所述之磁盘驱动器, 其特征在于, 所述扇区脉冲逻辑电路计算前述数据记录磁盘上的数据扇区的起始位置和数据扇区号。

45. 如权利要求 41 所述之磁盘驱动器, 其特征在于, 所述扇区脉冲逻辑电路接收一伺服模数计数, 此计数用于计算数据扇区的位置和特征。

46. 如权利要求 41 所述之磁盘驱动器, 其特征在于, 所述格式表包括用于前述数据记录磁盘上的每个存储段之子段以及存储段内每个数据区的存储项, 该存储项包含有用于上述存储段中数据扇区的位置及标识信息。

47. 如权利要求 40 所述之磁盘驱动器, 其特征在于, 所述伺服电子线路包括一数据扇区号计数器, 此计数器对相对索引的数据扇区号进行计数, 并且, 此计数器是在伺服扇区处预置好了的。

48. 如权利要求 47 所述之磁盘驱动器, 其特征在于, 预置所述的数据扇区号能避免存储带转换之后的等待。

49. 如权利要求 47 所述之磁盘驱动器, 其特征在于, 预置所述的数据扇区号计数器能避免在从节电模式返回之后的等待。

50. 如权利要求 47 所述之磁盘驱动器, 其特征在于, 所述的预置值提供了数据扇区与伺服扇区之间的错位。

51. 如权利要求 41 所述之磁盘驱动器, 其特征在于, 所述格式表包括用于每个存储带的存储项, 此存储项能使得前述伺服电子线路确定不紧接在伺服扇区之后的数据扇区的位置并标识该扇区。

52. 如权利要求 40 所述之磁盘驱动器, 其特征在于, 它还包括一伺服模数计数器, 此计数器能连续计算出存储段号和伺服扇区号,

这些号码对应于上述记录头相对数据记录磁盘的位置。

53. 如权利要求 52 所述之磁盘驱动器, 其特征在于, 对上述伺服模数计数器的输入包括交错偏移量。

54. 如权利要求 53 所述之磁盘驱动器, 其特征在于, 所述交错偏移量包括偏移值和时序调整值。

55. 如权利要求 54 所述之磁盘驱动器, 其特征在于, 根据磁头位移量将上述交错偏移量组织成一查寻表。

56. 如权利要求 40 所述之磁盘驱动器, 其特征在于, 调整所述的数据扇区号以补偿柱面和磁头错位。

57. 如权利要求 56 所述之磁盘驱动器, 其特征在于, 所述调整包括用一错位值修正所说的数据扇区号。

说明书

用于固定块磁盘驱动器的扇区结构

本发明在总体上涉及到数据记录磁盘驱动器,具体地说,本发明涉及到固定块结构(FBA)磁盘驱动器中所使用的多磁道数据存储介质的扇区格式。

本申请涉及到于1993年12月23日提交并转让给IBM的题为“数据记录磁盘驱动器内用于跳越扇区映象的系统与方法”的共同未决专利申请。

所有的磁盘驱动器均需要有某些用于确定读/写磁头在磁盘上径向及圆周位置的装置,以便使磁头能精确定位于任何希望的磁道及扇区。一般地说,这是靠在一个或多个磁盘表面上设置伺服信息来实现的,而磁头或光头则在确定自身在磁盘上位置时使用该伺服信息。在扇区伺服区(也即是嵌入伺服式)磁盘驱动器中,伺服信息与数据散布于每个磁盘表面上。这种方法的优点是能提供定位信息,该定位信息位于它所标识的数据扇区附近,因此,能够消除会限制磁道密度的磁道误登录的起因。但是,这种扇区伺服方法的缺点是:为进行数据区与伺服区之间的转换并把数据区与伺服区区分开来因而导致了额外的开销。

近年来,注意力集中在减少与扇区伺服结构有关的开销方面。Hetzler在1991年7月10日提出的共同未决的第07/727680号美国专利申请中说明了一种无标识格式的方法。无标识磁盘驱动器使用

与缺陷映射相配合的伺服扇区，从而标识出数据扇区并完全避免使用标识区。在使用上述无标识格式的情况下，一条磁道上的每个扇区都包括两部分：伺服区和数据区。通过使用伺服标识符或地址标记而使伺服扇区定位。可将错误校正码 (ECC) 加到磁道标识信息中从而提供更健全的伺服模式。每个数据扇区都是由其柱面、磁头以及从一索引位置算起的伺服扇区号所标识的。这种格式基本上对磁盘所有磁道上的全部扇区来说都是相同的。

正如 Hetzler 在第 5210660 号美国专利中所说明的那样，近年来用于增加扇区伺服式磁盘驱动器记录密度的第二种方法是区位记录 (ZBR) 法。在 ZBR 式磁盘驱动器中，磁盘被划分成多个沿径向方向排列的存储区。每个存储区均包括一组磁道。由于外部存储区上的磁道要比内部存储区上的磁道长，所以外部存储区上的磁道可以比内部存储区上的磁道存储更多的数据。一般地说，数据是存储在扇区中的，每个扇区均有着相同的数据字节数。在这种结构中，通过使外部存储区上的每个磁道具有更多个数据扇区从而能利用该外部存储区上的额外容量。这就导致每个磁道中数据扇区的数量会因所在存储区而异。为了能对所有的存储区都提供恒定的伺服取样率，因此在整个磁盘上使用唯一固定数量的伺服扇区。每一磁道上不定数量的数据扇区与每一磁道上固定数量的伺服扇区之间的组合，会导致某些数据扇区被伺服扇区所分割。图 1 显示了如 Hetzler 所说明的格式化了的磁盘的一个实例，其中，数据记录磁盘 101 被分成三个存储区 102, 104 和 106。每个存储区均包括有多条磁道 103。每条磁道上均具有若干个数据扇区 105 以及相关的标识字段 107。在整个磁盘中，数据扇区 105 分布于如标号 108 所示的多个伺服扇区之间。图中还

显示了索引位置 109, 在该位置处, 每一存储区内的数据扇区均与伺服扇区 108 相对齐。在标号 110 处以放大的方式显示出磁道的一部分。也显示出四个完整的数据扇区 (130、122、132 以及 124), 每个数据扇区均带有相关的标识字段 (140、141、142 和 143)。还显示出了三个示意性的伺服扇区 125、126 和 128。正如从这一实例中所看到的那样, 某些数据扇区被伺服扇区所分割, 而且, 某些数据扇区的起点并不紧接着伺服扇区。例如, 数据扇区 122 和 124 分别被伺服扇区 126 和 128 所分割, 而数据扇区 130 和 132 则未被伺服扇区所分割。数据扇区 122、132 和 124 以及相关的标识字段的起点紧接在另一数据扇区之后, 而不是紧接着伺服扇区。

必须将数据扇区分开并具有某些起点不紧接着伺服扇区的数据扇区是很复杂的, 这种复杂性妨碍了在同一磁盘驱动器内同时使用 ZBR 和无标识格式。例如, 在无标识磁盘驱动器中, 数据扇区的物理位置源于地址标记字段, 此地址标记字段也被用于确定伺服扇区的位置。但是, 这种技术依赖于伺服扇区位置与数据扇区位置之间固定不变的一对一关系, 这种关系在 ZBR 格式磁盘驱动器中是不存在的。

最近, 业已提出了一种技术, 这种技术通过设置电子线路以便产生时序脉冲去标记不一定与伺服扇区相邻且有可能被伺服扇区所分割的数据扇区的位置。从而解决了上述问题中的一部分。这种技术是 AT&T (美国电话电报公司) 在 ATT 93C010 伺服通道/多处理器芯片中提出的, 该芯片为每个起始于两个伺服扇区间的数据扇区产生一个数据扇区起始脉冲。这是通过使用其值在每个伺服扇区处均被更新的可编程寄存器而实现的。在每个伺服扇区处均需要有两个值:

从前一伺服扇区到第一个完整数据扇区起点的时钟脉冲数(长度)以及在下一个伺服扇区之前起始的数据扇区数。而且,系统必须知道一个完整的数据扇区所需要的时钟脉冲数,对每个存储区来说,该脉冲数的数值一般是一个常数。

但是,虽然 ATT93C010 可以确定数据扇区起始位置,却无法识别数据扇区,即通过诸如计算其数据扇区号的方法将一个数据扇区与其它数据扇区区分开来。事实上,ATT93C010 甚至不能计算局部的数据扇区号以便用于在磁道的同一部分上将一个扇区与其它扇区区分开来。因此,在不使用标识字段的情况下,无法将 ATT93C010 用于必须能同时确定数据扇区位置并识别该数据扇区的磁盘结构中。

对于在不使用标识信息的情况下确定数据扇区位置的另一种可能性是在每一数据扇区之前增加一伺服地址标记。这种办法能够确保以与伺服扇区无关的方式去确定每个数据扇区的位置。但是,这种方法有几方面的缺陷。首先,由于上述地址标记必须是由一个永久固定磁盘格式的伺服写装置写入的,所以上述方法允许将磁盘驱动器重新格式化具有与原有扇区大小不同的扇区大小。其次,由于需要附加的步骤去建立附加的地址标记,所以上述方法会增加伺服写处理的复杂性。第三,所述地址标记要占据磁盘空间,因而增加了开销。第四,所述地址标记需要在数据扇区之间有一写-读恢复区,这也增加了开销。最后,当和磁阻读/写头及微动技术一道使用时,必须在写操作过程中能将每一地址标记从部分偏离磁道的位置上可靠地读出。这就需要有位于所述存储区边界处的保护间隔,这是因为,所说的数据地址标记从一个存储区到下一个存储区并不依次排成一条直

线。当然，上述保护间隔也会增加与这种方法相关的开销代价。

因此，在本技术中，对能有效地将 ZBR 和无标识格式结合在一起的扇区结构来说，存在着至今尚无法满足的要求，上述扇区结构能够在不借助地址标记并不需要相邻数据扇区之间有写-读恢复区的情况下使数据记录磁头确定数据扇区的位置并识别该数据扇区，以便进行读写操作。

依照本发明，在数据记录磁盘驱动器中提供有 FBA 扇区结构，此结构能仅仅根据从电子存储器及不一定与数据扇区相邻的伺服扇区中所获得的信息使数据记录头确定所述数据扇区的位置并识别该扇区。上述磁盘驱动器中数据记录磁盘被径向划分成若干存储带 (zone)，每个存储带都包括若干磁道。每一磁道包括若干数据扇区和若干伺服扇区，数据扇区的数量因存储带而异。所述磁道包含伺服信息与数据，但不包含数据扇区的标识信息。在上述电子存储器中保存有格式信息以便说明用于每个存储带的的数据扇区格式。每个存储区中的磁道按圆周分成若干存储段。每个存储段内包括若干数据区，这些数据区被伺服扇区所彼此分隔。给定存储段内的所有数据区在长度上有着相同的字节数，但该长度不一定是数据扇区的偶数倍。因此，各个数据区均可包含以下任何一种内容：起点位于前一数据区内的数据扇区的终点；一个或多个完整的数据扇区以及终点位于下一个数据区内的数据扇区的起点。设置格式信息是为了识别包括数据区位置不与伺服扇区相邻的数据扇区在内的数据扇区，并确定该数据扇区的位置。该信息包括一用于存储段内每个数据区的登录项。每个登录项均包括：本区内第一个和最后一个数据段的长度；那些段所代表的扇区是否被分割于两个区之间；区内数据扇区的总数以及区

内第一个数据扇区的扇区号。除了记录用于每个区的信息以外,对于每个带,还保存有数据,用于确定起点不紧接于伺服扇区的数据扇区的位置。此数据包括:一完整数据扇区的字节数;一完整数据扇区的长度(用字节时钟来度量单位);一VCO(电压控制振荡器)同步字段的长度(以字节时钟来度量);位于伺服扇区之后的VCO再同步字段的长度(以字节时钟来度量);一个或多个数据填充字段的长度(以字节时钟来度量);以及每条磁道上数据扇区数与每条磁道上伺服扇区数比值化简后的分子和分母。数据区信息与存储带信息的一部分可存储于随机存取存储器中,该存储器可为磁盘驱动器中的电子线路所存取,其它部分则存储在一组寄存器中,该寄存器可为微处理器所存取。

依照本发明,还提供了一种方法,以便根据上述信息计算任一数据段的起始位置和扇区号。利用一组计数器去测算上述格式中各字段的长度,以便计算出所需数据扇区的起始位置。根据相关数据扇区号、相对索引的伺服扇区号以及每条磁道上数据扇区的化简数来计算出相对索引的数据扇区号。

依照本发明,还提供了一种方法和系统,以便连续地进行用于计算存储段号和伺服扇区号的伺服模数计数。所说的伺服模数计数器计算两个量:当前存储带的存储段结构信息的索引值;以及在当前存储段中第一个数据扇区的数据扇区号。上述伺服模数计数的连续有效性会使得具有本发明结构的磁盘驱动器避免时间等待的代价,这种等待的代价是因等待一索引标记而引起的,此标记是计算存储段号和伺服扇区号的基准。

依照本发明,还提供了一种方法和系统,以便识别在磁头转换之

后特别用于以交错伺服模式写入的磁盘驱动器的伺服扇区号。伺服扇区号的一部分写入每个伺服扇区并与从电子存储器中所获得的偏移值相配合,以便随磁头转换而产生伺服扇区号。

依照本发明,还提供了一种方法和系统,以便从伺服扇区号中计算出数据扇区号,这里,磁道上的第一个数据扇区与第一个伺服扇区之间的关系在一个存储带内以及从一个存储带到另一个存储带都将发生变化。这就使得本发明为磁盘驱动器提供了优点,这种磁盘驱动器被设计成带有最小化磁头和柱面转换操作等待时间的磁道错位。

图 1 是说明现有技术的固定扇区结构的概略图;

图 2 是说明本发明固定块结构磁盘驱动器的概略图;

图 3a 是说明本发明数据记录磁道的存储段子段的概略图;

图 3b 是说明本发明带有存根的磁道格式的概略图;

图 4 是说明本发明伺服功能的硬件实施例的框图;

图 5 是说明本发明用于计算数据扇区位置的存储器组件的框图;

图 6 是说明本发明伺服模数计算的流程图;

图 7 是说明本发明伺服模数计数器的硬件实施例的框图;

图 8 是说明交错扇区伺服写入磁盘驱动器的概略图;

图 9 是说明本发明 LBA (逻辑块地址) 至 PBA (物理块地址) 映射结构的概略图;

图 10 是说明本发明 LBA 至 PBA 转换计算的流程图;

图 11 是说明本发明存储带转换存储器组件的存储器映射表;

图 12 是说明从 PBA 至存储带、柱面、磁头、扇区的计算的流程图。

I. 扇区结构

图 2 显示了具有本发明之结构的磁盘驱动器。利用有扇区伺服和带位记录的固定块结构将该磁盘驱动器格式化。通常标号为 202 的磁盘驱动器包括: 数据记录磁盘 204; 活动臂 206; 数据记录变换器 208 (也称记录磁头); 音圈电机 210; 伺服电子线路 212; 读/写电子线路 213; 接口电子线路 214; 格式化电子线路 215; 微处理器 216 及 RAM (随机存取存储器) 217。数据记录磁盘 204 包括一旋转中心 211 并为磁头的定位而划分为一组径向间隔的磁道, 其中之一如 218 处所示。磁道被径向地分成若干的存储带, 其中的三个如 251、252 和 253 所示。磁盘含有多个伺服扇区 220, 这些扇区沿基本上为径向的方向跨越所说的磁道。每个磁道有一参考索引 221。在每个存储带内, 磁道按圆周方向被分成若干个数据扇区 254。正如下文将要说明的那样, 这些数据扇区不含有扇区标识字段。按照“固定块结构”的正常含义, 所有的数据扇区均基本上具有相同的以数据字节来表示的大小。但是, 应该注意, 如果对特定的实例来说是有必要的话, 那么就可以很容易地对本发明作出修改以容许数据扇区的大小在某些不同, 诸如从每个扇区 512 字节到每个扇区 520 字节。每一磁道的数据扇区的数量因存储带而异, 并且, 某些数据扇区的起点并不紧接着伺服扇区。此外, 伺服扇区将某些数据扇区分割开来。如果磁盘驱动器具有多个磁头, 那么就将在整个磁盘表面上处于相同半径处的磁道组称为一个“柱面”。

读/写电子线路 213 接收来自变换器 208 的信号且将伺服信息传递给伺服电子线路 212 并将数据信号传递给格式化器 215。伺服电子线路 212 利用所述的伺服信息去产生位于 240 处的电流。此电

流驱动音圈电机 210 以使记录变换器 208 定位。接口线路 214 通过接口 262 与一主机系统 (未显示) 相通讯并传递数据和命令信息。接口线路 214 还通过接口 264 与格式化线路 215 相通讯。微处理器 216 则通过接口 270 与其它各个电子线路相通讯。

在磁盘驱动器 202 的运转过程中, 接口线路 214 通过接口 262 接收一读或写数据扇区的请求。格式化线路 215 从接口线路 214 中接收被请求的数据扇区表并将它们转换成存储带、柱面、磁头及数据扇区号, 它们能唯一地确定所期望的数据扇区的位置。磁头与柱面信息传递给伺服线路 212, 该线路负责使记录头 208 在适当柱面中适当数据扇区上定位。如果提供给伺服线路 212 的柱面号与记录头 208 当前被定位的磁道号不相同, 那么, 伺服线路 212 就会先进行查找操作, 以便使记录头 208 重定位于适当的柱面上。

一旦伺服线路 212 使记录头 208 定位于适当的柱面上, 伺服线路 212 就开始扇区计算操作以定位并识别所期的扇区。当伺服扇区 220 在记录头 208 下方通过时, 可使用上述美国专利申请第 07/737680 号中的无标识方法去识别每个伺服扇区。简要地说, 索引标记识别出第一个伺服扇区, 地址标记确定出后续伺服扇区的位置, 而地址标记计数则唯一地识别出每个伺服扇区。与伺服电路 212 和格式化电路 215 一同维护的其它信息, 在后面将对其进行更详细的叙述。该信息用于确定当前伺服扇区是否把一个数据扇区分割开来了, 或者新的数据扇区是否紧随在当前伺服扇区之后开始。在伺服线路 212 和格式化线路 215 内还保存有其它信息, 此信息能确定出下一个数据扇区相对当前伺服扇区的起始位置 (或距离)。还保存有另一些信息, 该信息能确定出在下一个后续伺服扇区之前开始的任何额

外数据扇区的位置(或距离)。再有一些信息能识别出相对索引标记的数据扇区号。该信息用于格式化电子线路 215 将在记录头下面通过的数据扇区号与从接口线路 214 接收到的扇区表进行比较。

图 3a 显示了取自本发明之数据记录磁盘的示意性磁道的扇区结构的详细图。磁道的一部分如 302 所示,它包含有存储段 304。存储段 304 细分成多个数据段 306、308 和 309。这些数据段被伺服扇区 310、312 和 314 所彼此分开。存储段 304 还包括标号为 01 至 05 的多个数据扇区。最后,每个数据扇区均由一个或多个数据段构成,这些数据段标号为 320、322、324、326、328、330 和 332。从逻辑上说,一个存储段是伺服扇区与数据扇区的集合,这些伺服扇区与数据扇区之间有着唯一的空间关系。通过重复上述存储段就形成了磁道的格式。数据区是位于相邻伺服扇区之间的空间。数据扇区则是最小的逐个可寻址的用户数据单元,该数据单元可独立地进行读和写。最后,数据段是数据扇区的邻接部分,此部分未被伺服扇区所隔断。

图 3a 中的示意性磁道包含有数量不一定相等的多个数据扇区和多个伺服扇区。应该注意,由于数据扇区 D5 刚好在伺服扇区 316 之前结束,所以伺服扇区 316 不是所述存储段的组成部分。每个数据区均包含有多个数据扇区,其中有些扇区被伺服扇区所分开。例如,数据区 306 包含完整的数据扇区 D1(数据段 320)以及数据扇区 D2(数据段 322)的一部分。同样地,伺服扇区 312 将数据扇区 D2 分成数据段 322 和 324。

图 3a 还显示了伺服扇区 310 的详细内容。写-读恢复和速度补偿字段 342 用于使读/写电子线路从数据写操作转换至伺服读操作并用于处理磁盘旋转速度的波动。地址标记字段 344 精确地标识出

伺服扇区内的某一特定位置，此位置被用作定时的基准。位置字段 346 包含有用于使磁头定位的具体伺服信息，它典型地包括一位置错误信号以及诸如磁道号（磁道标识或 TID）、索引值、伺服扇区号（或该扇区号的一部分）以及磁头号（或该磁头号的一部分）。伺服填充字段 348 允许伺服线路从读取伺服转换至写或读数据，并且允许磁盘旋转速度变化。

图 3a 也显示了包含有整个数据扇区 D5 的数据段 332 的详细内容。VCO 同步字段 352 允许读/写电子线路让电压控制振荡器（也称锁相环路）获得用于读取数据的适当锁相。数据与 EEC 字段 354 包含有用户数据和错误校正信息。数据填充字段 356 供读和写数据的处理时间存在差异时使用，也供清除任一编码器/译码器时使用，同时还供磁盘旋转速度发生变化时使用。该字段还提供了足够的时间以使读/写电子线路作好处理下一伺服扇区或数据扇区的准备。

图 3a 亦显示了标号为 360 的被分割开的数据扇区 D2 的详细图。当伺服扇区将数据扇区分割开来时，一般需要有两个附加字段：分割填充字段 364 和 VCO 再同步字段 368。字段 322 表示数据扇区 D2 在伺服扇区 312 之前的那一部分。分割填充字段 364 供读/写电子线路以本技术中众所周知的方式中断读或写数据。伺服扇区 312 的后面是 VCO 同步字段 368，该字段用于存储系统以供连续进行读或写操作。最后，在 324 处显示了数据段 D2 位于伺服扇区 312 之后的那一部分。应该注意，分割填充字段 364 可以在长度上与数据填充字段 356 有着相同字节数，也可以有所不同。此外，VCO 再同步字段 368 在内容上可与 VCO 同步字段 352 相同，但并不要求一定是这样。通过使字段 364 和 368 短于相应的字段 352 和 356，可以获得更

大的容量;在以往技术中说明了用于获得这种收益的方法。

对任何给定的数据记录磁盘驱动器来说,整个磁盘中每条磁道上都有着固定数量的伺服扇区(以下记为 N)。而且,对每个存储带来说,每条磁道上都有着固定数量的数据扇区(以下记为 M)。如果 M 不是 N 的倍数,则就会有某些数据扇区被伺服扇区分开。每个数据扇区的被分开部分称之为数据段。此外,属于数据扇区的第一个数据段称为主数据段,而其余任何段则均称为从数据段。由于磁道上所有的数据扇区有相同的字节数并且伺服扇区是等距间隔的,所以,磁盘驱动器上存在着数量有限的独立数据段。定义数据段独立结构一个周期的数据扇区和伺服扇区的集合称为存储段。一个存储段上数据段的数量(以下记为 nss)是由下式给出的:

$$nss = m + n - 1 \quad (1)$$

其中, $\frac{m}{n}$ 表示 $\frac{M}{N}$ 的化简分数。

由于 m 和 n 表示每条磁道上数据扇区数与每条磁道上伺服扇区数之比的化简后分数的分子和分母,所以,很明显,一个存储段上有 n 个伺服扇区和 m 个数据扇区。对图3a中所示的示意性存储段304来说, $n = 3$ 、 $m = 5$ 、 $nss = 7$ 、 $N = 84$ 、 $M = 140$ 以及 nst (每条磁道上存储段的数量) = 28。应该注意,按照无标识扇区结构,伺服扇区和数据扇区都不包括数据标识字段。相反,正如以下将要详细说明的那样,在伺服扇区310、312、314等以及能为伺服电子线路所存取的电子存储器中提供有识别数据扇区号和数据扇区位置所必需的信息。

应该注意,可以灵活地选择存储段结构。例如,可将整个的磁道定义为一个存储段。在某些情况下,诸如当 M 和 N 互素因而 $m = M$, $n = N$ 时,这可能是很自然的选择。但是,不可能排除将 $\frac{m}{n}$ 选择为化

筒后分数的整数倍的情况。而且,存在着这样的情况:对定义一个存储段来说,上述有关伺服扇区数与数据扇区数之比的分析并不是最佳的选择。当在磁道终点处剩余有空间,即该磁道上最后一个数据扇区基本上在该磁道的终点之前结束时,上述情况就会发生。图 3b 说明了这种情况。磁道的标号为 380 并包含有 7 个伺服扇区 384。有 11 个数据扇区 DS1 至 DS11。应该注意,数据扇区 DS11 在磁道 380 终点之前结束。剩余空间是剩余部分 385,由于该剩余部分要比数据扇区短,所以它不包含用户数据。在本例中, $N=7$ 和 $M=11$,根据上述分析,这将产生 $n=7$ 和 $m=11$ 的存储段大小。但是,图 3b 还说明了第二种可能性。当 $n=3$ 和 $M=5$ 时可以实现伺服扇区与数据扇区之间的唯一空间关系。这种选择会形成两个完整的存储段 381 和 382 以及一个不完整的存储段 383。磁盘驱动器只需知道磁道上数据扇区的总数,就能处理上述不完整的存储段。一旦数据扇区号达到了最大值,驱动器就会等待下一个伺服扇区,该伺服扇区将数据扇区计数器恢复至第一个数据扇区号。应该注意,可以选择具有多个剩余部分的磁道格式,其中包括位于磁道内以及位于磁道终点处的剩余部分。总之,通过用一较小的存储段去重定义数据区,有时在存在剩余部分的情况下可以获得某种优点,这是因为,这会减少存储上述格式信息所需的存储量。

II. 格式化及伺服电子线路

图 4 是本发明之伺服及格式化电子线路的最佳实施例的概略图,伺服及格式化电子线路用于确定数据扇区的位置并识别该扇区。伺服电子线路 212 包括:地址标记 (AM) 探测器和磁道号 (TID) 译码器 412; 伺服扇区计数器 414; 保险逻辑电路 416; 运动器位置控制逻

辑电路 418; 时间发生逻辑电路 420; 扇区脉冲发生逻辑电路 421 以及格式表 422。格式化电子线路 215 包括: 缺陷映射表 450、数据扇区识别器 454; 逻辑块地址 (LBA) 到物理块地址 (PBA) 的转换器 456; PBA 至存储带、柱面、磁头和数据扇区 (ZCHS) 的转换器 458 以及控制功能电路 476。

在运行过程中, 格式化线路 212 接收用于对数据扇区表 264 作读或写操作的请求。扇区的 LBA 标识出所说的扇区。转换器 456 利用缺陷信息 460 将 LBA 表转换成 PBA 表。PBA 表 468 再被转换成物理 ZCHS 表。在以下第 VI 部分中将详细说明上述转换过程。将柱面和磁头值 466 (C 和 H) 传给启动器位置控制逻辑电路 418 以便进行一次查寻操作。而启动器位置控制逻辑电路 418 以本技术中众所周知的方式起作用。将存储带和扇区值 464 (Z 和 S) 传给数据扇区识别器 454 和伺服电子线路 212。此外, 伺服电子线路 212 接收来自读/写电子线路的伺服信息 266。AM 探测器 412 探测伺服地址标记并在 432 处发出地址标识找到 (AMF) 信号。此信号传给时序逻辑电路 420, 该电路会产生伺服电子线路 212 所需的时间信号。AMF 432 还被传给伺服扇区计数器 414。另外, AM 探测器 412 译解包括柱面(磁道)号、伺服索引、伺服扇区号和磁头号在内的 TID 信息。索引信号 433 用于复位伺服扇区计数器 414, 该计数器在每次后续的伺服时因 AMF 信号 432 而增值。通过这种方式, 伺服扇区计数器总是在 436 处输出当前的伺服扇区号。保险逻辑电路 416 接收已译解的 TID 信息 430 和伺服扇区号 436。此逻辑电路进行各种保险测试以确保伺服电子线路的正确操作。其功能包括: 将所产生的伺服扇区号 436 与 TID (包括索引) 中的任一伺服扇区号信号相比较; 以及处理 TID 中

的误操作信息。在 438 处输出有错误信息以及柱面号和磁道号。启动器位置控制电路 418 将柱面及磁头值 438 与目标值 466 作比较并作用于任何误差。扇区脉冲逻辑电路 421 利用伺服扇区号 436 生成用于格式表 422 的地址, 以便检索用于存储带的存储段信息 440 (以下将详细说明)。扇区脉冲逻辑电路 421 还包含有三个计数器: 同步计数器、数据计数器以及填充计数器。这些计数器中的每一个均用来对上述各式中各字段内的字节时钟脉冲进行计数, 据此而确定数据扇区的位置并识别该扇区。因而, 扇区脉冲逻辑电路 421 可以识别上述扇区的起始位置以及将要在记录头 208 下通过的数据扇区的扇区号。当前的数据扇区号 422 被送至格式化电子线路 215, 并且, 在数据扇区的起点位于磁头下方时传送数据扇区开始脉冲 444, 从而使得格式化线路实现具有零等待的功能。一旦接收到扇区脉冲 444, 数据扇区识别器 454 就会将当前的数据扇区号 442 与 ZCHS 值的表 464 作比较。如果找到匹配值, 该信息就会经由 470 传给控制功能逻辑电路 476, 在这一位置, 按以往技术中已知的方法来对数据扇区作进一步处理。

参照前述字段, 应该注意, 存在有多种能达到上述目的的替代结构。例如, 可用任何逻辑标识来代替 LBA, 并可用能标识一个唯一扇区号的一个值或几个值的组合来代替 ZCHS。上述电子线路与微处理器 216 中微码相配合可以检测到并处理在目标 ZCHS 值与所检测到的值之间所显示出的任何错误。即使在没有数据标识字段的情况下, 这也为所说的磁盘驱动器提供了高度的可靠性。例如, 如果所检测到的磁道号与目标值不匹配, 则会登记下查寻错误, 并如在本技术中已知的那样去处理该错误。保险逻辑电路 416 将检测诸如伺服扇

区计数器与 TID 信息 (索引以及可能的伺服扇区号数位) 之间不匹配之类的其它错误, 以便由所述驱动器的其它部分来加以处理。这样的错误可以当作数据 ID 误比较来处理, 并且通过强制伺服扇区计数器与索引标记相匹配并重复该操作来进行恢复。最后, 如以下将要详细说明的那样, 伺服模数计数是扇区脉冲逻辑电路 421 连续计算数据扇区位置所需的重要输入。该伺服模数计数通过利用格式中的重复模式而减少该格式信息所需的总存储量。

III. 确定数据扇区的位置及识别数据扇区

图 5 显示的是本发明之扇区计算所需的组件。总的来说, 提供了一随机存取存储器 504 和一组标号为 506 的寄存器, 它们均与地址和数据总线 (未显示) 相连接。这些组件可驻留在伺服电子线路 212 内, 在格式表 422 中。微处理器 216 (或其它电子线路) 将信息存储于随机存取存储器与寄存器中, 然后再存取这些信息, 以便进行扇区识别及定位计算。

具体地说, RAM 504 存储有识别给定独立存储段中每一个数据段内的数据所需的信息。尽管可以使用任何适当的数据结构, 但是, 为便于说明起见, 将图中所示的信息字段组织成了表的格式。通过所述存储段内的数据段 (相当于该存储段内的伺服扇区号) 去访问 RAM 504。每个数据段所需的字段包括 DS1Len 508、DS1Num 510、DS1V512、NumFull 514、DS2Len 516 以及 DS2V 518。DS1Num 是伺服扇区之后第一个数据扇区自存储段起点处算起的号码。在所述的最佳实施例中, 该字段为一个 7 位的值。DS1Len 是所述数据区内第一个数据段的字节长度。在所述最佳实施例中, 该字段含一个 10 位的值。DS2Len 是所述数据区内最后一个数据段的字节长度。在所述

的最佳实施例中, 该字段为一个 10 位的值。NumFull 是所述数据区内完整(未被分割的)数据扇区的数量。在所述的最佳实施例中, 该字段为一 3 位的值。DS1V 是一标识, 如果设置该标识, 则表示第一个数据扇区是被分割开的, 因此 DS1Len 的值是有效的。在所述的最佳实施例中, 该字段为一个 1 位的值。

由于给定的存储带内的各个存储段均有相同的字节数, 所以, 对每个存储带来说, 只要存储一个存储段的格式信息就可以了。存储段内的伺服扇区号(等同于数据区号)用来访问适当存储段的格式信息。包括 DS1Num 值的目的是为了提提高效率。扇区脉冲逻辑电路 421 包括一数据扇区号计数器, 该计数器从索引开始对数据扇区脉冲进行计数。只要伺服电子线路保持着活动状态, 数据扇区号计数器就会是正确的。但是, 当要进行存储带转换时, 就必须预置该数据扇区号计数器。另一方面, 最好避免等待用于开始对数据扇区计数的索引。这是因为, 这会导致等待时间的代价。利用 DS1Num 值, 在每个伺服扇区处预置计数器, 从而避免了额外的等待。此外, 这种技术能使磁盘驱动器迅速从节电状态复原, 在节电状态下, 多数电子线路在不执行数据操作时都会降低电力消耗。利用本发明, 可在节电结束之后的第一个伺服扇区处而不是在索引处将数据扇区计数器预置成正确的值。

在图 3a 所示的扇区结构的定义中, 任何给定的数据区内只有第一个和最后一个数据段会包含不完整的扇区。所以, 只需将这两个片的长度存储于所说的表内, 从而包括了 DS1Len 和 DS2Len。NumFull 是所述数据区内完整数据扇区的数量。它用于防止在数据区结束之前产生错误的的数据扇区脉冲, 这是通过一旦产生了

NumFull 个数据扇区脉冲就使数据扇区号计数器不能工作来完成的, 在数据区的终点处, 数据扇区的终点可能靠近伺服扇区的起点。NumFull 还用来控制产生数据扇区脉冲, 此脉冲用于起始于所述数据区内的数据扇区。DS1V 和 DS2V 用于指示 DS1Len 和 DS2Len 值的有效性。这是由以下原因引起的: 必须知道数据区中的第一个数据段是否为主数据段从而使初始化于一个伺服扇区的读/写操作不会开始于一个从数据段。例如, 在数据区 308 中, 伺服扇区 312 之后的 DS1Num 值是用于数据段 324 的 D2。但是, 对数据扇区 D2 的读或写请求必须始于数据段 322, 所以, 伺服电子线路必须确保不会将数据段 324 误识别为数据扇区 D2 的起点。与此相似, DS2V 用于指示最后一个数据段被伺服扇区所分开。由于上述标识值提供了直接译码, 所以使用这些值对提高效率来说是最佳的。但是, 很明显, DS1Len 和 DS2Len 值的有限范围允许使用特定的值 (如 0) 以提供同样的功能。

只须将一数据区中的第一个和最后一个数据段长度存储于表内, 这是因为, 其它所有数据段都一定具有完整的长度。由于所有其它的数据段都一定是完整的, 所以只有第一个和最后一个数据段才需要数据段分割标识。如果第一个数据段是主数据段 (即未被分割), 则 DS1V 为零。同样, 如果最后一个数据段是被分割的, 就用 DS2V 在该数据段结束处执行分割数据扇区功能。

应该注意, 如图 5 所示, 可将上述信息存储于 RAM 存储器中并通过存储段内的伺服扇区号来访问这些信息。但是, 也可以用数据扇区号来形成上述信息, 而扇区号则可用于访问 RAM。在这种情况下, 必须要更改特定的字段, 但这些字段仍会支持前述功能。此外, 也可

将以上信息中的一种或全部存储于其它方便的介质中，如存储于寄存器、瞬时存储器或其它可为伺服电子线路 212 所存储的存储设备。

除了上述存储在存储段中每一数据区内的信息以外，对每个存储带来说还需要其它信息以便使伺服处理器确定未紧接在伺服扇区之后定位的数据扇区的位置。在图 5 中，这一信息通常包含在寄存器 506 中，这些寄存器包括 Sync Long 寄存器 520、Sync Short 寄存器 522、Pad Long 寄存器 524、Pad Short 寄存器 526，以及 Data Full 寄存器 528。寄存器 520 - 528 用于装载扇区脉冲逻辑电路 421 中的三个计数器。这三个计数器（同步计数器、数据计数器以及填充计数器）都是递减计数器，其输入的时钟脉冲是当前的数据字节时钟脉冲。按着磁道格式所确定的先后次序一次只能有一个计数器工作。用于其它操作的辅助信息包含在 n 寄存器 530、m 寄存器 532、NDS 寄存器以及 DSKew 寄存器 536 中，n 寄存器 530 存储有每条磁道上数据扇区的化简数，m 寄存器 532 存储有每条磁道上伺服扇区的化简数，NDS 寄存器 534 存储有每条磁道上数据扇区的数量，DSKew 寄存器 536 则存储有数据扇区的误差值。

具体就寄存器 520 - 528 而言，Sync Long 寄存器 520 包含有 VCO 同步字段 352 内的字节时钟脉冲数。Sync Short 寄存器 522 包含有 VCO 再同步字段 368 内的字节时钟脉冲数。在所述的最佳实施例中，字段 368 要比字段 352 短，但是，如果这两个字段是等长的，则只需要有 Sync Long 寄存器 520。当数据填充字段 356 的后面是 VCO 同步字段 352 时，Pad Long 寄存器 524 包含有上述数据填充字段内字节时钟脉冲数。当数据填充字段 356 的后面是伺服扇区时，Pad Short 寄存器 526 包含有上述数据填充字段内的字节时钟脉冲

数。Pad Long 字段与 Pad Short 字段的值是不同的，这是因为，所说的伺服扇区包含着 W-R 与速度字段 342，该字段与数据填充字段 356 有着共同的功能。Data Full 寄存器 528 包含有数据与 ECC 字段 354 内的字节时钟脉冲数，也即一数据扇区内数据与 ECC 字节的总数。

在伺服电子线路 212 与 RAM 504 和寄存器 506 相互配合的操作过程中，同步计数器在伺服扇区之后启动。在这之前，如果 DS1V 的值是 0，则从 Sync Long 寄存器 520 中预装入该同步计数器，而如果 DS1V 的值是 1，则从 Sync Short 寄存器 522 中预装入该同步计数器。此外，如果 DS1V 的值是 0，则产生数据扇区脉冲 444，并将 Data Full 寄存器 528 内的值预装入数据计数器。否则将 DS1Len 的值装入该数据计数器。当同步计数器到达 0 时，就启动数据计数器。如果在所述数据区内所产生的数据扇区脉冲的数量等于 NumFull，则将 Pad Short 寄存器 526 的值装入填充计数器，否则将 Pad Long 寄存器 524 的值装入该填充计数器。当数据扇区计数器到 0 时，就启动填充计数器。当该填充计数器到 0 时，就已到了当前数据扇区的终点。在数据区内所产生的数据扇区脉冲的数量等于 NumFull，则紧跟着的就是一个伺服扇区。否则，就产生一数据扇区脉冲，数据扇区号计数器加 1，并将 Sync Long 寄存器 520 的值预装入同步计数器。重复这一过程直至数据区结束，而这一点则是由于产生了 NumFull 个数据扇区脉冲所决定的。当在数据区内所产生的数据扇区脉冲的数量等于 NumFull 且 DS2V 为 1 时，则将 DS2V 的值而不是 Data Full 寄存器 528 的值预装入数据计数器。

用三个计数器来代替一个计数器是因为要对多种字段进行计

数，以便限制单个计数器中分接点的数量以及在其它计数器运行时预装入每个计数器。应该注意，虽然公开了特定最佳计数器的结构，但也可以使用在本发明的精神和范围内的多种其它计数器和寄存器来实现上述功能。

通过利用上述信息，伺服电子线路 212 可以确定一个存储段中任一数据扇区的起始位置。此外，可以利用下述等式来确定来自于给定数据段的索引的扇区号：

$$SN(i) = m \times SGN + DS1Num + i \quad (2)$$

其中， $SN(i)$ 是来自于存储段中第 i 个数据扇区的索引（以零为基点）的数据扇区号， SGN 是来自索引（0 为基点）的存储段号。

在磁盘驱动器 202 的运行过程中，收到读写操作，该操作要求记录头 208 在各磁道上重新定位然后读或写各数据扇区。一旦记录头 208 定位于适当的磁道上，就可用上述等式确定适当的数据扇区何时在记录头下经过。具体地说，用该等式来计算当前的数据扇区号，此数据扇区号则不断地与读或写所请求的数据扇区号作比较。如果两者的值相等，则进行所希望的操作。由于没有标识字段，所以，以上述数据值为基础计算出的数据段长度会确定特定存储段与数据区内数据的位置。

在所述的最佳实施例中，上述信息字段中的每一个都保存在各存储区域中，以便提高操作效率。例如，将用于当前存储带的信息保存在与寄存器 520 - 536 相类似的专用局部存储器中以避免总线判优。在每个存储带转换之后可从诸如 RAM 217 之类的通用存储器中重新装入所说的值。磁盘的格式决定着用于特定实例而存储上述值所需要的存储量。由于已知的实例需要每个存储带有 32 个或更少的

数据区,所以存储带表的局部存储需求应为每个数据区 30 位或每个存储带 256 字节,包括于用 ECC 的容量。因此,在所述的最佳实施例中,可在没有存储器存取代价的情况下取得较高的效率。

IV. 伺服模数计数器

为了有效地利用上述扇区识别/定位方案,本发明的系统必须能在执行读或写操作之前确定存储段内的存储段号和伺服扇区号。为了避免等待作为计算上述信息基准的索引而引起的等待时间代价,设置了伺服模数计数器电子线路以便即使在磁头或存储带转换之后也能不间断地产生所说的信息。在所述的最佳实施例中,这是通过硬件电路而实现的,硬件电路的输入是 m 、 n 、Servo (伺服扇区数) 以及 DS1Num。

伺服模数计数器的目的是产生 $\text{Servo mod } n$ 及 $m \times (\text{Servo}/n)$ 量。前者是 Servo/n 的余数,即相对存储段起点的第 n 个伺服扇区的号码,它被用作前述以 RAM 为基准的存储带表 504 的地址。后者是所述存储段内第一个数据扇区的数据扇区号,即数据扇区的基准号,以下记为 DSB,它也是等式 (2) 中的第一项。图 6 以流程图的方式说明了进行这些计算所需的操作。在 602 处将余数值赋为伺服计数的值。在 604 处将 DSB 的值赋为 0。在 606 处进入循环,此循环通过重复的减法直至产生溢出而计算 $\text{Servo mod } n$ 的值(余数)。具体地说,在 606 处将余数 $-n$ ($\text{Servo}/n - n$) 的值赋给临时变量 temp。在 608 处进行下溢测试 ($\text{temp} < 0$)。如果测试到下溢,则在 610 处退出处理过程。如果没有测试到下溢,则在 612 处继续处理,在 612 处,将 m 的值加至 DSB。最后,在 614 处使余数等于 temp (即余数减 n),处理过程返回至 606。在出口 610 处,已经计算出了 $\text{DSB}(m \times (\text{Servo}/n))$

和余数 ($\text{Servo mod } n$)。

图 7 显示了说明本发明之最佳实施例的电路的概略图。该电路按照图 6 所示之流程图的过程及上述方法来计算 $m \times (\text{Servo}/n)$ 和 $\text{Servo mod } n$ 的值。如该流程图所示, 通过重复的减法来计算 $\text{Servo mod } n$, 而减法则由一下溢操作来终止。电路包括控制逻辑 702、减法器 704、加法器 706、寄存器 708 和 710、多路转换器 (MUX) 712、加法器 714 以及数据扇区号计数器 716。在该伺服模数计数器电路的操作过程中, 控制逻辑 702 接收起始信号 720 以便开始计算并在计算结束时产生完成信号 744。寄存器 708 存储重复减法的结果并在计算结束时包含着 $\text{Servo mod } n$ (余数) 的正确值。减法器 704 接收来自寄存器 708 的输入 738 以及来自寄存器 530 (先前参照图 5 对此寄存器作了显示和说明) 的输入 (值 n)。寄存器 708 的输入来自 MUX 712。该 MUX 将所述的分子、Servo 728 或先前减法的结果装入寄存器 708。寄存器装载信号 (未显示) 来自控制逻辑 702 并在每次减法操作之后出现。下溢信号 742 则传给控制逻辑 702。如果测试到下溢, 寄存器装载信号就中止且余数值将稳定地存在于 740 上。控制逻辑 702 还控制经由选择线 732 从 MUX 712 到寄存器 708 的输出。MUX 712 具有这样的结构, 它能使第一次作减法时使用当前的伺服计数 728, 而后续的减法则使用中间结果。此外, 由于退出条件是下溢, 寄存器 708 位于减法器 704 之前, 以便该寄存器在循环结束时包含有先前减法的结果。

对在取模部分所进行的每次减法操作来说, 通过反复增加 $\text{Servo mod } m$ 的值而计算出 $m \times (\text{Servo}/n)$ 的值。加法器 706 增加来自寄存器 532 (先前参照图 5 对此寄存器作了显示和说明) 的输入 730 (值

m) 与先前加法的结果 746。该加法的中间值 748 在加法器 706 输出时存储于寄存器 710 中。寄存器装载信号 (未显示) 来自控制逻辑 702 并在每次加法操作之后出现。控制逻辑 702 通过上述寄存器装载信号使加法和减法操作同步, 从而排除了需要有单独的乘法器或累加器来计算 $DSB(m \times (Servo/n))$ 。当上述操作完成时, 按等式 (2) 用由上述电路所产生的 DSB 值 746 去算出当前的数据扇区号。具体地说, 结果 DSB746 和 DS1Num754 一道传给加法器 714, 而 DS1Num754 则是由余数 740 所确定的。这些值被加在一起并作为从存储段中第一个数据扇区索引处算起的数据扇区号在 750 处输出。一旦从控制逻辑电路中收到装载信号 (未显示), 就立即将加法器的值 750 预装入数据扇区数计数器 716。数据扇区数计数器 716 在接收到数据扇区脉冲 756 时增加并在 752 处输出当前的扇区号。

为了避免因磁头或磁道转换引起的等待代价, 上述转换起因于确定新磁道所需的时间, 本技术中周知的是使用数据记录磁盘驱动器的磁盘表面与磁道之间的柱面和磁头错位。

依照本发明, 提供了一种技术, 通过调整数据扇区计数器 716 内的值来解决错位问题, 从而实现错位。通过从当前数据扇区号 752 中减去 Dskew 寄存器 536 的值, 调整原始的 (未错位) 的当前数据扇区号 752。可以通过对 NDS 寄存器 534 中的值取模来执行上述减法。其结果是当前已错位的数据扇区号, 此扇区号可作为输入 442 提供给数据扇区识别器 454。最后, 应该注意, 虽然参照数据扇区说明了所说的错位技术, 但是, 也可以同样有效地使用这种技术以与上述相类似的方式在伺服扇区号的基础上实现错位。

V. 交错扇区伺服

对磁头和存储区转换来说,按照最佳实施例,使用基于硬件的扇区模数计数器能保证不会增加等待时间。对存储带转换而言,伺服扇区计数器保持伺服位置的计数。一旦改变了 m 和 n 的伺服与数据模值,数据段长度和数据扇区号也将修正。与此相似,对磁头转换而言,一旦改变了伺服计数,其余的值也随之改变。

对实现了交错扇区伺服方法的磁盘驱动器来说,必须随着磁头转换而适当地设置伺服计数器的值,以便解决交错。图 8 以剖面图的方式显示了交错扇区伺服磁盘驱动器。通常磁盘驱动器的标号为 802 并包括主轴 804 和磁盘 806、808 及 810。用两组黑色矩形 820 和 822 概略地显示了伺服扇区的位置。这些伺服扇区不是垂直对齐的,相反,它们是以斜列方式排列的,从而能以每条磁道单遍写入的方式来对磁盘驱动器的所有表面作伺服写操作。在使用一个磁头写某一表面上的伺服扇区之后,启动(下一个表面上的)下一个磁头并在该处写伺服扇区,从而在一次旋转过程中写多个表面,这就减少了伺服写的时间和成本。

为了写本发明的交错扇区伺服模式,伺服扇区计数器必须与所转换到的表面上的伺服扇区号相同步。可通过将伺服扇区号的一个子集(也可能是整个的扇区号)写入位置字段 346 的 TID 信息中来实现这一功能。另外,可以根据一查寻表来更新伺服扇区计数器 414,查寻表包含有交错偏移量。在前一种情况中,从伺服扇区中读出的伺服扇区号被 AM 检测器 412 所译解并传给保险逻辑电路 416。保险逻辑电路 416 用这个值去更新伺服扇区计数器 414。

例如,可将整个的伺服扇区号经编码放入 TID。磁头转换之后,保险逻辑电路 416 用上述读出的值去预装伺服扇区计数器 414,从

而保证了同步。在使用查寻表的情况下, 备有偏移量表, 该表用于增加或减少伺服扇区计数器 414 内的值。840 处显示了一个示意性的表。该表包含三列, 即磁头位移量 842, 伺服扇区计数器偏移量 844 以及伺服时序调节量 846。一旦开始进行磁头转换, 就用磁头位移量的值 (加号为向下、减号为向上) 去查寻伺服扇区计数器偏移量和时序调节量。偏移量的值 844 用于使伺服扇区计数器 414 增值或减值。而时序逻辑电路 420 则使用时序调节量的值去调整新伺服扇区的位置。为方便起见, 将列 846 中的值列表为伺服对伺服间隔的比值形式。因此, 例如, 如果当前磁头位于磁盘 806 的下表面上, 并且, 希望磁头转换到磁盘 810 的上表面上, 则磁头转换值应该为 +3。从查寻表 840 中可知, 伺服扇区计数的增值为 0, 伺服扇区时序调节量应为伺服对伺服间隔的 $1/2$ 。图 8 所示的表结构也可用于诸如使索引从一个表面错位至另一个表面之类的更一般的偏移量。这就将导致每个磁头位移值具有唯一值。此外, 也可以结合对 TID 内伺服扇区号编码来使用所说的表, 从而增加系统的可靠性。当然, 此表可以存储于 RAM 或其它适当的介质中。

VI. 从 LBA 转换至 ZCHS

如前所述, 为了在磁盘上找到所请求的数据扇区, 必须把已收到的逻辑块地址 (LBA) 转换成存储带、柱面、磁头及扇区 (ZCHS) 的值。一般地说, 这个过程包括首先将 LBA 转换成物理块地址 (PBA), LBA 是用于数据扇区的用户标识, 而 PBA 则是 LBA 至磁盘驱动器的映像。前述 Hetzler 的申请 (07/727680) 说明了 LBA 至 PBA 映射的缺陷映射及其基本过程。但是, 用于相邻缺陷的 LBA 占用了大多数高位数位, 所以, 存储在缺陷映射中的多数信息都是多余的, 因而

只起妨碍效率及多余占用 RAM 的作用。本发明包括了特定的映射，此映射除去了每个 LBA 的高位数位，从而将表示各个缺陷所需的数位减至最少。上述映像通过使用最少量的 RAM 并以提高磁盘驱动器效率的方式顾及位于磁盘上由缺陷或冗余所导致的跳跃扇区。

上述映射是通过一对表来实现的，即一个虚拟磁道 (VT) 表和一个虚拟扇区 (VS) 表，虚拟磁道表包含有表示多数 LBA 之冗余的高位数位的记录项，而虚拟扇区表则包含有仅表示 LBA 之低位数位的记录项。然后，将 VT/VS 表存取的输出即 PAB 放至一存储带变换表内，从而求出用于计算柱面、磁头和扇区的因数，以便对所请求的扇区进行所希望的操作。因此，要使用两种变换，一种是移去跳跃扇区，另一种是提供适当的 ZCHS 值。本发明将转换所需的存储量减至以下大小：每记录项两个或一个字节再加上一个基于磁盘容量的偏移量及一个或两个字节记录项。本发明还会减少确定 LBA 位置所需的检索次数，从而减少了对上述检索过程效率的影响。

图 9 显示了 LBA 至 PBA 的映射结构，该结构构成了所述变换过程中 LBA 至 PBA 变换部分的基础。具体地说，该图显示了 904 处的以二进制形式表示的 LBA、虚拟磁道表 906、虚拟扇区表 908 以及缺陷备用表 910。如 904 处所示，从磁盘驱动器接口接收来的 LBA 首先被分成两段。如 A 所示的 LBA 的以位表示的全部长度包括：高位部分上存储虚拟磁道号所需的若干位，记为 B；以及存储虚拟扇区号所需的若干位，记为 C。

将虚拟磁道定义为一个精确地具有 $2^{(A-B)}$ 个好数据扇区的数据扇区的相邻集。虚拟磁道可以包含好扇区和跳跃扇区，或者只包含好扇区。所有的虚拟磁道从所定义的零号虚拟磁道开始都是彼此相邻

的并遍及整个磁盘驱动器，零号虚拟磁道则位于磁盘驱动器的起始处。也可以同样的方式单独映射磁盘驱动器的子空间。可以通过单纯选择 LBA 的高位数位或者将 LBA 右移位 $A - B$ 位而从 LBA 中获得虚拟磁道号。将虚拟扇区定义为包含在虚拟磁道之内的好扇区。存在有 2^{A-B} 个可以得到的虚拟扇区号。因此，虚拟磁道内的所有虚拟扇区被顺序编号，始于磁道内第一个好扇区、止于磁道内最后一个扇区。

如果将 LBA 细分成虚拟磁道号和虚拟扇区号，则将虚拟磁道号加入到虚拟磁道表 906，以便得到一索引点进入虚拟扇区表 908。索引点被用作 VS 表 908 的起始点，在此起始点处开始进行顺序检索，这种检索操作一直持续至比所要检索的虚拟扇区号大的虚拟扇区号位于虚拟扇区表 908 内或所说的索引超过用于下一个虚拟磁道的记录项。一旦找到了适当的记录项目，将所接收到的 LBA 加上与最后一个检索记录项相对应的进入 VS 表 908 的索引来算出 PBA。

VT 表 906 包含有与磁盘驱动器中每条虚拟磁道相对应的记录项。每个记录项至少为 B 位宽并包括含有指向 VS 表 908 的指针。因此，该表的结构使得每个记录项具有两层含义。第一，记录项的值标识着该记录项所代表的虚拟磁道之前的跳跃数据扇区的总数。第二，该记录项的值还代表进入虚拟扇区表的正确入口点，在所述虚拟扇区表中列出所说的跳跃扇区以用于上述记录项所代表的虚拟扇区号。在操作过程中，根据从记录项 904 中所获得的 B 值即虚拟磁道号作索引而进入 VT 表 906。然后，将在确定的索引点处所找到的值加到 VS 表 908 的起始处以便确定与可以使用的虚拟磁道相关的第一个跳跃扇区的位置。

与 VT 表 906 不同, VS 表 908 不包含用于磁盘驱动器上每个虚拟扇区的记录项。相反, VS 表 908 只包含用于跳跃扇区(缺陷扇区和/或备用的第一个扇区)的记录项。VS 表 908 内的每个记录项均与跳跃扇区之后的好的虚拟扇区相对应。如上所述,可通过用 VT 表 906 所建立的索引值进入 VS 表 908。根据接收自 LBA 值 904 低位的 C 位,从上述索引值处开始进行顺序检索,直至找出一虚拟扇区号,此扇区号大于从 904 的高位 C 位所获得的值。该更大的扇区号建立起跳跃次数,经过上述若干次跳跃到达正确的虚拟扇区号。如果下一个虚拟磁道不是从索引所在的位置开始的,在这种情况下将偏移量/索引加到 PBA 的值中,那么,就将等于虚拟扇区表 908 的第一个记录项内的偏移量的偏移值加至 LBA 以便获得 PBA 的值,而所述的第一个记录项要大于所要查找的虚拟扇区号。

DS 表 910 包含着用于 VS 表 908 中各个记录项的记录项。DS 表 910 的目的是为了区别 VS 表 908 内的虚拟扇区记录项与缺陷数据扇区即坏扇区相对应还是与备用的数据扇区即备用块相对应。在所述的最佳实施例中,DS 表 910 包含一个一位的记录项,此记录项与 VS 表 908 的每个记录项相对应,其中,所述各个记录项中该数位的相反性质用于区别坏块和备用块。应该注意,DS 表 910 的优点在于:除了重新赋值操作过程以外,该表不必驻留在 RAM 存储器中。在使用磁盘驱动器的过程中形成了逐渐增加的缺陷时,就会产生重新赋值操作,这时,将备用扇区重新指定为活动扇区,并将上述有缺陷的活动扇区重新指定为跳跃扇区。重新赋值操作包括改变 DS 表 910 内的数位值以及 VT 表 906 和 VS 表 908 内的各个值。当然,为了能改变 DS 表 910 内的值,该表必须驻留在能为上述微处理器所存

取的 RAM 中。而在其它情况下, DS 表 910 不必是常驻的或准备为磁盘驱动器电子线路所存取, 这是因为, 不需要由上述电子线路来确定 VS 表 908 内所标识的扇区是坏的还是备用的, 相反, 该电子线路仅需确定扇区是跳跃扇区。

图 10 显示了说明 LBA 至 PBA 转换计算的流程图。该变换计算过程具有三个基本功能。首先, 设置将要通过 VS 表 908 进行的检索操作的边界。这是通过检验 LBA 中所标识的虚拟磁道号记录项以及 VT 表 906 中后续虚拟磁道号记录项以形成虚拟扇区表中的起始和终止偏移量而实现的。其次, 一旦设置了所说的检索边界, 上述变换过程就会顺序检索 VS 表中的 VS 号, 直至该表中的 VS 的位置等于下一个虚拟磁道的起始点(即上述检索操作的上界)或者找到大于 C 位的 VS 表记录项。第三, 作为上述检索操作的标准, 虚拟扇区表 908 中的每个虚拟扇区号必须大于或等于前一个扇区号, 除非是超越了虚拟磁道的边界, 而在这种情况下, 检索操作会终止。

具体参照图 10, 变量 D 表示 $A - B$ 的值, 其中, A 等于 LBA 的位数, B 等于虚拟磁道号的位数。因此, D 为获得虚拟磁道号所需的 LBA 向右移位数。变量 E 表示 $2^C - 1$ 的值, 其中, C 等于虚拟扇区号的位数。因此, E 只是为确定虚拟扇区号而提供了“与”掩码。应该注意, D 和 E 的值是根据系统参数而形成的。例如, 由于多数现代微处理器均能处理为 8 的倍数的数位, 所以最好形成 D 和 E 以便能以 8 位的倍数的方式提供所有的值。最后, P 是虚拟扇区表中的当前偏移量。

在 1004 处进入图 10 的流程, 其中, 将 VTN 设置成等于向右移位 D 位的 LBA。其次, 在 1006 处, 将虚拟扇区号设置成等于 LBA 加

上E。因此,在这一位置上就已经将逻辑块地址分解成了虚拟磁道号和虚拟扇区号。再次,在步骤1008处,将虚拟扇区表中的当前偏移量设置成等于在虚拟磁道号偏移量处所找到的虚拟磁道表中的值。这就形成一起始点以便进一步检索虚拟扇区表。然后,在步骤1010处,将E值设置成下一个虚拟磁道号(要对虚拟扇区表所执行的检索操作的上界)。

在1012处进入一循环,在此循环中,进行测试以确定是否到达了虚拟磁道的终点。应该注意,如果在所述循环的第一次重复中就产生上述情况,则结果表示当前磁道不包含有缺陷扇区。如果还未达到磁道的终点,则在1014进行第二次测试,在这次测试中确定被检索的虚拟扇区是否大于虚拟扇区表中所测试到的虚拟扇区记录项。如果测试结果是正的,这表明还未找到所要查找的索引。在这种情况下,执行步骤1016以便使虚拟扇区表中的当前偏移量增加到下一个虚拟扇区表的记录项。此后,控制返回1012处的循环起点。如果在1014处所进行的虚拟扇区号比较的结果是负的,这就表示在虚拟扇区表中已经找到所要查找的索引。对流程控制的效果(如同在1012探测到磁道的终点,并且进入1018,在1018处通过使PBA等于LBA加上虚拟扇区表中当前偏移量的值而结束检索操作。最后在1020处退出所说的计算过程。

通过使用虚拟磁道号的某些部分与虚拟扇区号之间的重叠,可以获得重要的优点。参照分别定义虚拟磁道号和虚拟扇区号的B位和C位,可从图9中904所示的LBA定义中看出上述重叠的特点。当出现这种重叠时,虚拟扇区号的一部分相当于虚拟磁道号的最低位。这种重叠是有意提供的,以便提高从LBA至PBA的映射效率。

虚拟磁道号与虚拟扇区号之间的重叠位能使得虚拟扇区在不必返回到虚拟磁道表的情况下就能区别相邻的和附近的虚拟磁道号。虚拟磁道号和虚拟扇区号之间的重叠量决定了用单个虚拟扇区号来彼此区别开的虚拟磁道的数量。由于虚拟扇区表中的记录项目是按递增数序排列的,所以,只要在能用单个虚拟扇区号区别开的虚拟磁道数的范围内存在有至少一个出现在虚拟扇区表中的跳跃扇区,那么,就可只使用虚拟扇区表来检测在虚拟磁道号中的所有变化并对和该表中各个记录项相关的虚拟磁道号进行计数。因此,在每次越过虚拟磁道边界时,不必再次访问虚拟磁道表就可以对虚拟扇区表进行检索。

在计算跨越磁道的跳跃之间的绝对距离时,将某些虚拟磁道号的最低有效位编入虚拟扇区号的第二个优点是很明显的。对由一个或多个磁道交叉所分开的两个跳跃来说,只要它们之间的距离处于虚拟扇区中重叠位数的分辨率范围内,那么就可从其它虚拟扇区值中减去用于跳跃扇区的虚拟扇区值,以求出上述跳跃之间扇区的绝对数量。因此,在仍不涉及虚拟磁道表的情况下就能跨越磁道边界来执行绝对距离的计算。

一旦计算出了物理块地址,就进行第二次变换,在这次变换中,将PBA进一步转换成存储带、柱面、磁头和扇区位置。图11说明了用于在带位记录磁盘驱动器中初始化上述变换过程所需要的存储带变换存储器的组成。该表的标号通常为1104并最好存储于RAM中,也可以存储于其它方便的介质中。存储带表1104包括三列,即存储带PBA列1110,存储带柱面列1112以及存储带扇区列1113。存储带表1104包括一用于磁盘驱动器内每个存储带的记录项。每个记录项的第一列即存储带PBA1110表示存储带的起始PBA,而且该列还

用于确定所需要的 PBA 位于哪一个存储带内。因此, 存储带表内的最上面的记录项表示磁盘驱动器的存储带 1 中的第一个 PBA, 等等。将所要查找的 PBA 号与每个存储带 PBA 记录项相比较, 直至找到包含该 PBA 的存储带。然后, 用存储带柱面列 1112 和存储带扇区列 1113 内相应的记录项目来进行转换计算, 以下将说明该转换计算。

一旦确定了存储带变换存储器的组成, 图 12 以流程图的方式说明了从 PBA 变换到 ZCHS 的步骤。在步骤 1204 处。在存储带表中查找所要寻找的 PBA 号以检索图 11 所述的存储带 PBA、存储带柱面以及存储带扇区的记录项。在步骤 1206 处, 从 PBA 中减去存储带 PBA 以确定在要找的 PBA 所驻留的存储带内有多少物理块 (扇区)。这步操作返回进入存储带的 PBA 偏移量。在 1208 处, 用每条磁道的扇区数除 PBA 偏移量。这步操作的模数或余数为数据传递的起始扇区号。在步骤 1210, 将上述除法的商取作相对存储带起点的磁道偏移量, 而数据的传递则起始于上述存储区起点。在 1212 处, 通过用磁盘驱动器内磁头数量除磁道偏移量以形成柱面偏移量, 然后该柱面偏移量被加到存储带的起始柱面号上, 从而计算出开始进行数据传递的绝对柱面号。最后, 在 1214 处, 将在 1212 处同一除法所提供的余数或模数取作用于开始数据传递的选定磁头。这里, 在已计算出开始数据传递的存储带、柱面、磁头和扇区的情况下, 在 1216 处退出所说的计算过程。

将 LBA 至 ZCHS 转换过程分解成两个阶段的一个好处在于两步映射过程, 此过程能显著地减少备用单元的数量, 而备用单元则必须被分配到整个磁盘驱动器以处理字段中产生的缺陷。从 LBA 至

ZCHS 过程的两个步骤转过来又将跳跃扇区的分配分解为一个两步过程。首先,对整个磁盘驱动器进行面分析测试(SAT)以便在从厂家发运磁盘驱动器之前确定工厂的缺陷的位置。通过按数序插入跳跃扇区而使上述缺陷映射进虚拟扇区表。而且,创建一虚拟磁道表,并且,如使位于其中检测到错误的磁道之后的每个虚拟磁道记录项增值所需要的那样来更新上述虚拟磁道表。其次,在 SAT 完成之后,备用存储单元就会按与已有缺陷的概率相一致的距离均匀地分布在整个磁盘上。应该注意,已有缺陷的概率以及备用单独的密度随磁盘上位置的函数而变化。其结果是将备用单元减至最低水平。

对上述用于备用单元的方案来说,提供了两种可替换的用于重新分配扇区的实施例。这两种重新分配技术均将备用扇区映射进磁盘驱动器的逻辑地址空间并将缺陷扇区映射进虚拟扇区表,并且,如果必要的话,还将上述缺陷扇区映射进虚拟磁道表,以便将该缺陷扇区指定为跳跃扇区。

对每一种技术来说,通过识别最近的可用备用扇区(用作重新分配的扇区)来开始上述重新分配过程。这包括存取 DS 表并对最接近上述缺陷的记录项作前后检索直至找到一备用记录项。从缺陷位置中减去前后记录项的位置,并且用最低的绝对值结果去选择重新分配的扇区。一旦确定了重新分配的扇区,就应用上述两种重新分配技术中的一种。

回到开头,原始表未作修改并保存有重新分配扇区的单独列表。此单独列表包含有从上述分布列表中选出的扇区,而分布列表则是当作重新分配的扇区而使用的。所说的列表呈映射的形式,该映射包括一第一列和相应的第二列,第一列表示现存有缺陷的 ZCHS 值,

第二列则表示新的 ZCHS 值。上述列表本身可由 LBA 或 ZCHS 号来组织。如果该列表由 LBA 号构成, 则在以 LBA 转换成 PBA 之前检测该列表。在这种情况下, 对上述列表的记录项加以解释以确定映射哪些 LBA 以及将这些 LBA 映射成什么样的新 LBA。如果所述列表是按 ZCHS 值组织的, 则在从 LBA 转换成 PBA 以及从 PBA 转换成 ZCHS 之后处理该列表, 并且对该列表的记录项加以解释以确定映射哪些 ZCHS 值以及将这些 ZCHS 值映射成什么样的新 ZCHS。总之, 当在数据传递操作过程中遇到缺陷重新分配扇区时, 就对该重新分配的备用单元进行第二次查寻, 并在该单元处读或写数据。但是, 如上所述, 由于所说的备用单元遍布整个的磁盘驱动器并且选择了该驱动器以便利用最靠近上述缺陷重新分配扇区的分布备用单元, 所以查寻的长度是最短的并因此而提高了效率。

第二种重新分配技术能有效地利用在虚拟磁道表和虚拟扇区表中所选定的重新分配备用单元来取代现存缺陷备用单元。通过在现存缺陷处插入一新跳跃扇区, 然后移动上述已有缺陷和重新分配的扇区之间的所有值, 就可以修改虚拟扇区表。如果最近的备用单元从缺陷处跨越磁道边界, 则更新虚拟磁道表以便指示出指针的改变。因此, 仅通过移动虚拟扇区表中所述缺陷与重新分配的扇区之间的 LBA, 就能使上述备用单元占据所说缺陷的逻辑位置并在此后不会降低效率。

应该注意, 上述跳跃扇区转换方案无须改进就在存在有错位的情况下具有额外的优异操作性能。在存在有错位的地方, 可将计算扇区号的先有计算方法看作是涉及到未错位的扇区。然后如前所述, 在上述转换过程之外转换至错位的扇区。

对上述在希望简化映射情况下可能是有效的实施例的一种改进是使用单个的表转换过程，此过程直接从 LBA 映射至 ZCHS。上述表的记录项同在现有技术中一样仍需要三或四个字节，但是，其新的表结构是以 LBA 为基础的，而该 LBA 则由表的位置所调整。因此，所说的表包含着 LBA，这些 LBA 紧接着一给定的跳跃。这一方法的优点在于：表的查寻过程只需要简单的一级检索，该检索为顺序的或是二进制的。只要系统内的跳跃的数量很少，则 RAM 的需要量是相当少的。但是，对新式大型磁盘驱动器来说，存储量需求会随缺陷的增加而迅速增长，从而无法使用这种方法。

还应注意，本部分所公开的跳跃扇区映射技术在无标识式磁盘驱动器范围之外也是有效的。事实上，可以将这种技术应用于任何磁盘驱动器，在这些磁盘驱动器中，由于在每次查寻操作开始时确实能确认最终的目标存储区、柱面和扇区，因而从中获得优点。所以，在所有的磁盘驱动器，包括那些使用标识字段以及使用无标识信息的磁盘驱动器，所公开的 LBA 到 ZCHS 映射会避免为处理备用空间及缺陷所需的磁头滑动。而且，所说的系统会很容易地实现适时的查寻开工文件，其中，使用精确查寻目标的资料来确定查寻路线，以便使磁头刚好在所需的扇区到达该磁头之前落在磁道上。最后，所说的系统还用于实现等待时间最少的查寻记录，这是因为在进行查寻之前就已知了准确的查寻目标。

更为重要的是，应该注意，虽然以上的说明重点在于将 LBA 划分成两部分，但是，其思想很容易扩展到三个或更多的划分部分。例如，可以加上一虚拟柱面部分，此部分包含着相邻的虚拟磁道，这在虚拟磁道数变大时是有优点的，这是因为它减少了用于虚拟磁道

表的存储量。因此,总的来说,上述映射方法应该看成跳跃扇区的 n 层层次存储结构。

Ⅶ. 结论

所述最佳实施例硬件结构的另一优点在于其内置式电源管理装置。由于所述的伺服-模数计数块实际计算模数,所以系统只需要单一的用于初始化的伺服扇区时间。扇区脉冲逻辑电路 421 中的字段长度计数器仅在读写模式过程中操作,否则就处于节电状态。因此,在本发明的低功率实例中,伺服电子线路 212 和格式化电子线路 215 的大部分都在读写操作之间处于节电状态。一般地说,时序逻辑电路 420 在这段时间是活动的并产生信号以激活 AM 探测电路 412 和每个伺服扇区的其它电路。由于伺服扇区计数器是活动的,所以在下一个伺服扇区之后就可以进行读或写操作。如果希望节约较多的电源,就可以增加一阶段,在这一阶段中,伺服扇区计数器和相关的电子线路均处于节电状态。在这种方式中,如果未将整个伺服扇区号编码进 TID,系统就必须等待一索引以初始化自己。如果已将整个的伺服扇区号作了编码,系统则只需等待下一个伺服扇区。

虽然是参照了数据扇区对本发明作了说明,而所说的数据扇区要比分割伺服扇区的数据区短,但应该注意,可以将本发明同样地应用于相反情况,即选定扇区的大小或伺服空间以便使数据扇区的长度大于数据区的长度。这种现象会出现于因效率或其它缘故而使用密集伺服扇区的实例中。这种现象也会出现于这种情况中:将数据扇区的大小定得相对较大,以便在应用中处理诸如多媒体图像及科学数据之类的长块信息。数据扇区的大小超过数据区大小的结果是一个数据扇区会跨越两个或多个伺服扇区。依照前述的本发明,通过相

应地简单编排寄存器和 RAM 值,就能进行上述处理。

尽管上述最佳实施例的硬件结构使用了一个扇区模数计数器以减少存储量,但是,很明显,可以扩大所述的存储段信息以存储用于整个磁道的信息。在这种情况下,由于通过伺服扇区号可直接访问 RAM504,所以不再需要伺服扇区计数器。这种结构比所述的最佳实施例使用更多的 RAM,但具有简化的硬件结构。对不同结构的选择取决于诸如用于两种装置的电路成本等因素。而且,可以将 RAM504 中的数据组织成能用除伺服扇区号以外的值来编址。具体地说是可以用数据扇区号来访问所说的表。

此外,应该注意,在某些实例中,从误差处理角度来说,在每个数据扇区内包括 LBA 的某些部分有好处。这一点在读操作过程中可用于作 LBA 验证,但在写操作过程中却是无用的,这是因为,所要查找的扇区已被重写了而不是被读取了。相应地,数据扇区内的 LBA 编码不同于数据扇区内的标识信息编码。而且,将如前所述的标识信息排除在外会实现一种实例,该实例从数据扇区信息中排除了所说的标识,但包括 LBA。

很明显,上述电子线路中所示的功能元件的位置是可以改变的,而且,在不脱离本发明精神与范围的情况下,也可以用软件来代替所述的硬件。更为明显的是,尽管是在存储带记录格式的范围内说明了上述最佳实施例,但是,本发明也可应用于其它的复杂格式,在这些格式中,只要该格式的内容对磁盘驱动器来说是已知的,磁道上的数据扇区数就可以不等于磁道上的伺服扇区数。而且,寄存器和 RAM 这些术语应被看作是可以相互交换的,并且寄存器和 RAM 的位置是无关紧要的。例如,寄存器 506 和 RAM504 可以在 RAM 217 而不

是格式表 422 中。最后, 应该注意, 虽然在磁盘驱动器的操作过程中将以上所述的值和表均存储于 RAM 及寄存器内, 但是, 在驱动器不操作时必须将它们存储于非易失性存储器中。可以使用任意的非易失性存储器, 但最好将所说的信息存储到磁盘驱动器自身。使用可修改的存储器(即所说的磁盘驱动器)可以使得该磁盘驱动器在不需外界干涉的情况下改变自身的格式。

尽管是参照一最佳实施例来说明了本发明, 但是, 本技术的专家应该认识到, 在不脱离本发明精神和范围的情况下, 可以在形式和内容方面对前述说明作出修改。因此, 应该将本发明看作是包括了在下述权利要求所限定范围内的所有改进形式和变化形式。

说明书附图

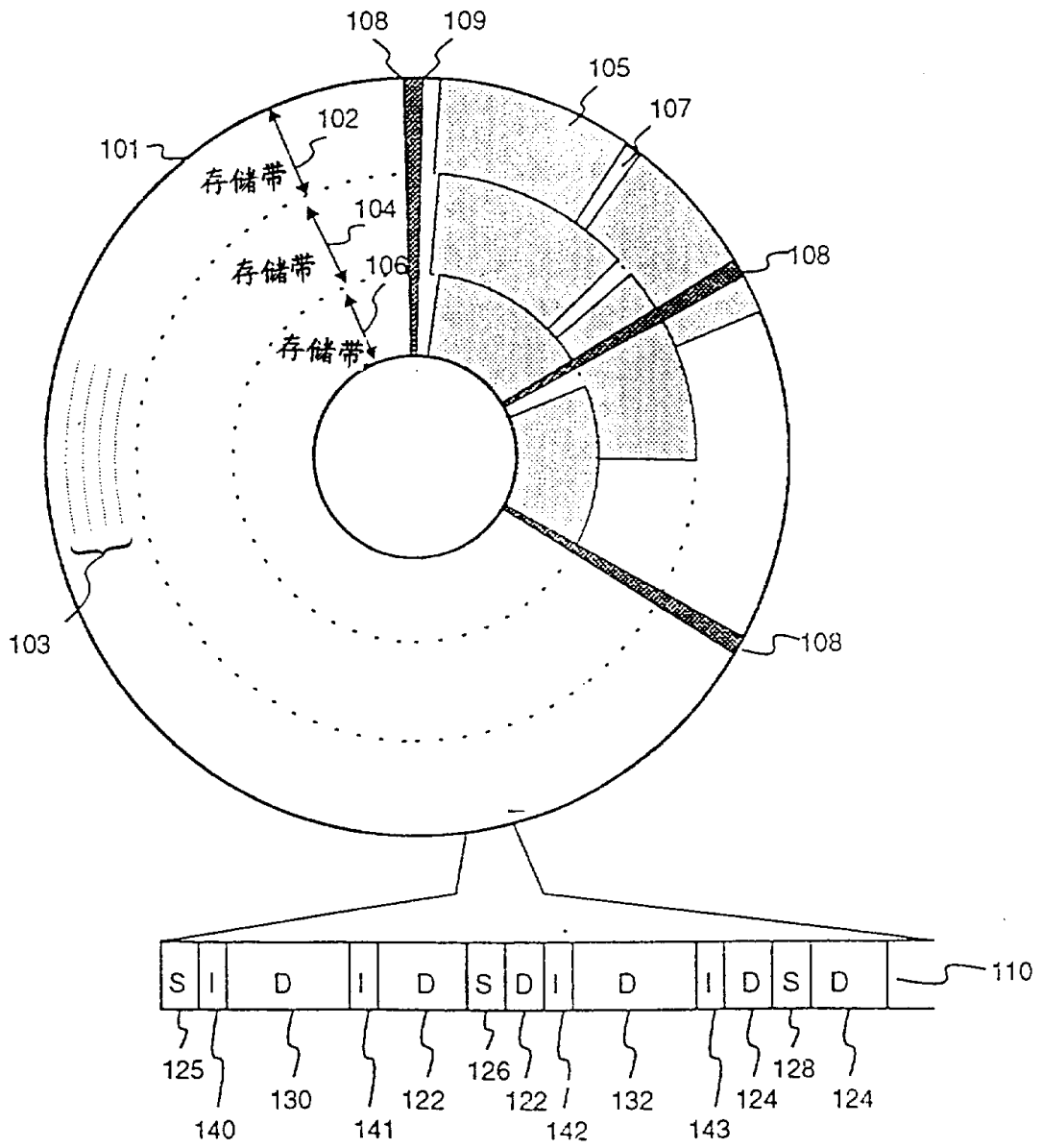


图 1 现有技术的扇区结构

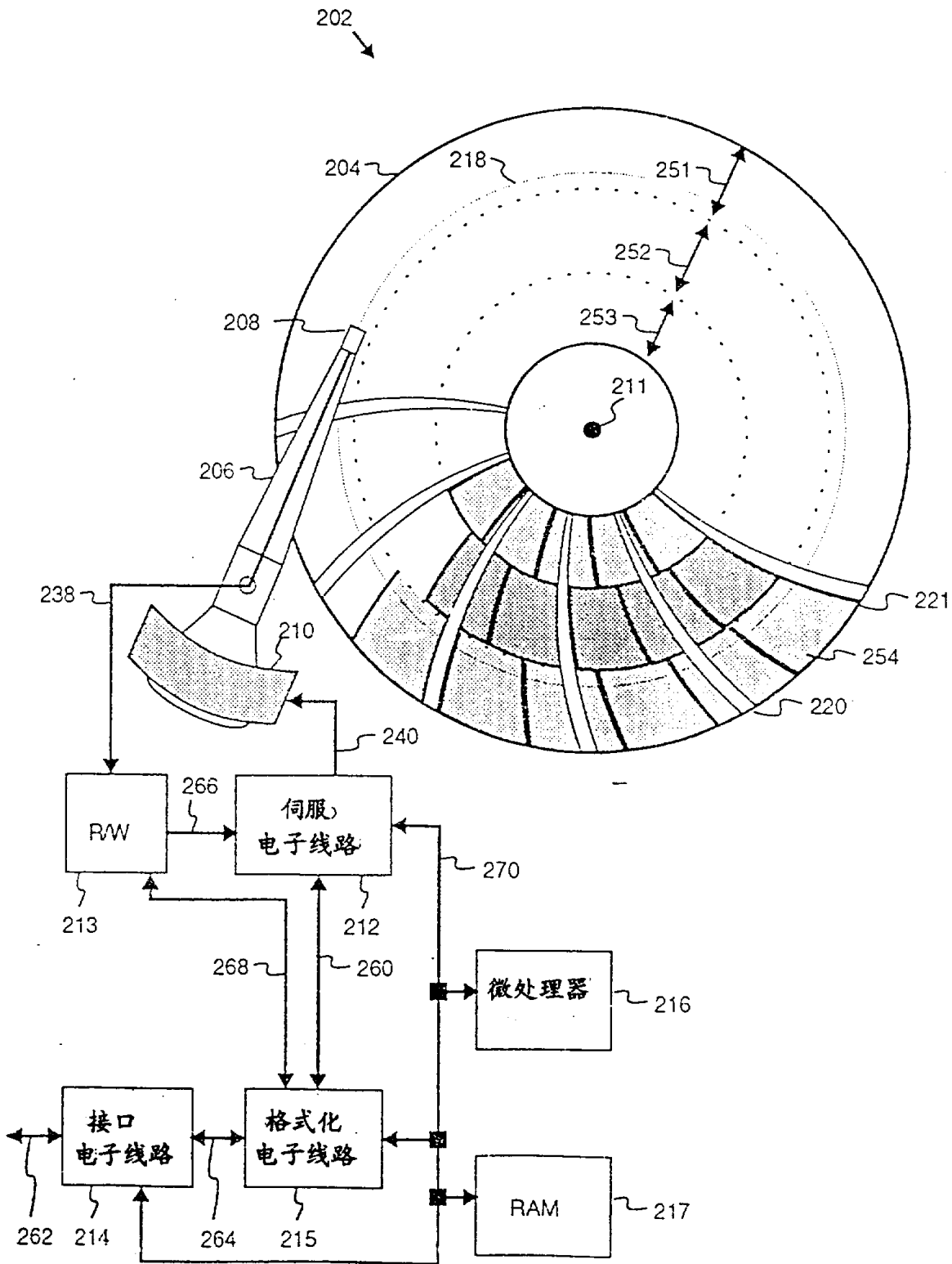


图2 磁盘驱动器

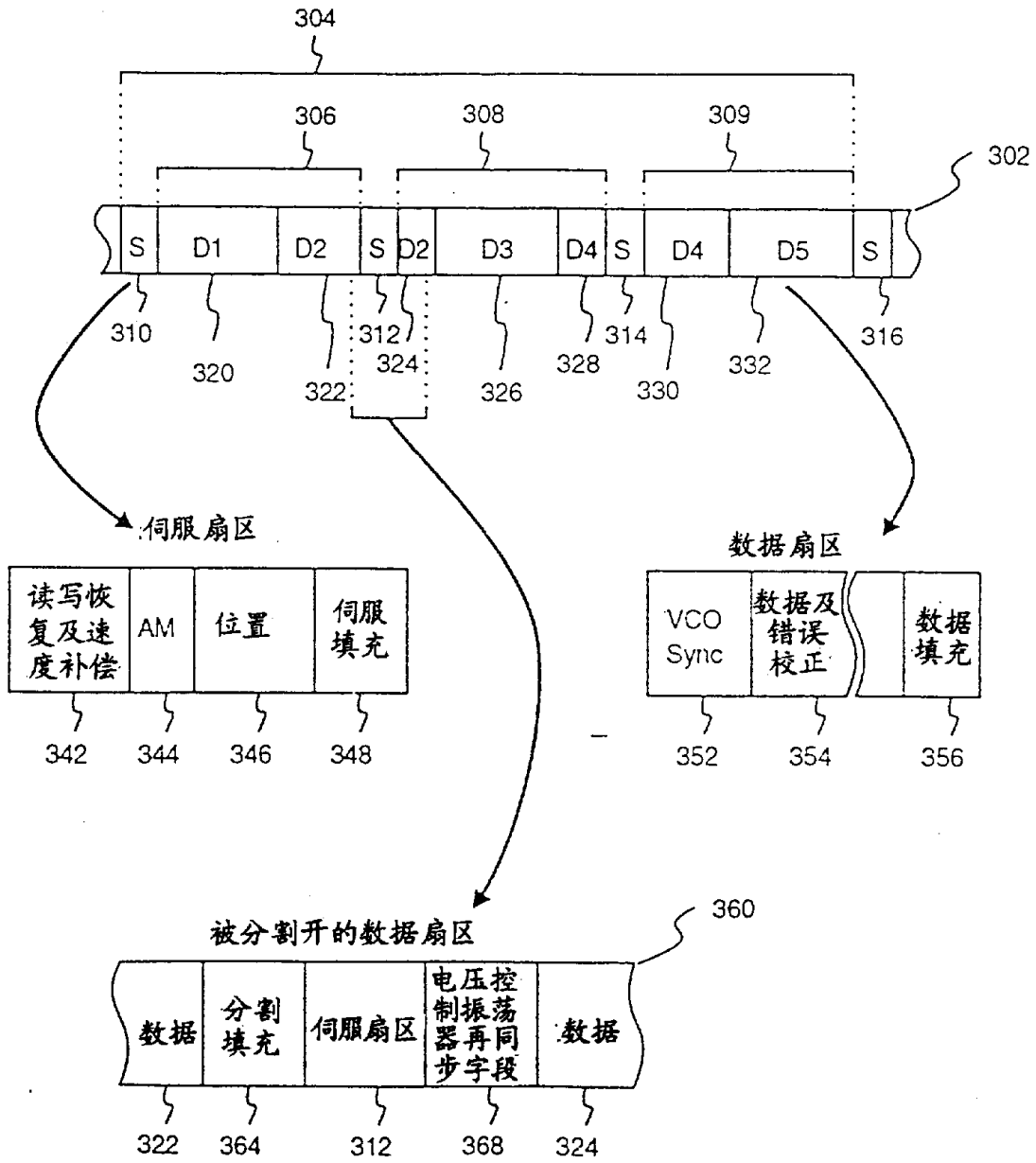


图 3 a 扇区结构

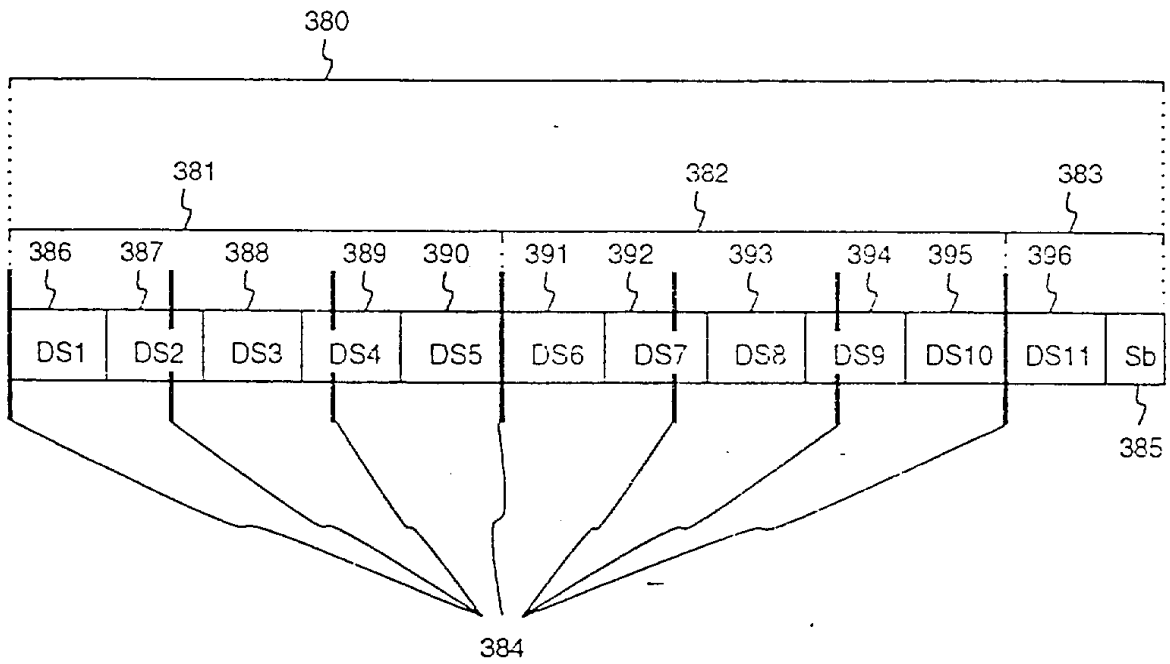


图3b 带有存根的磁道格式

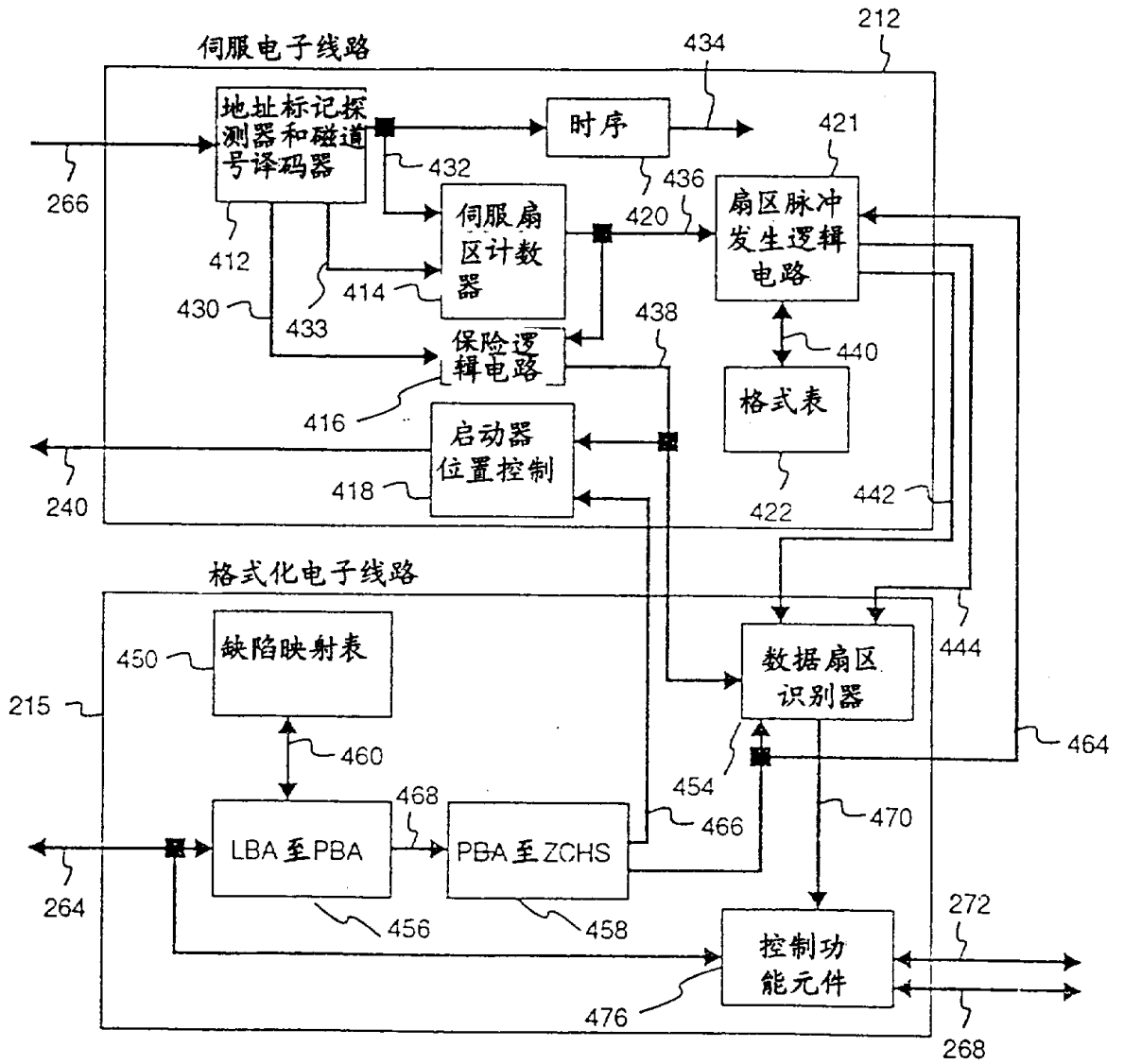


图 4 电子线路

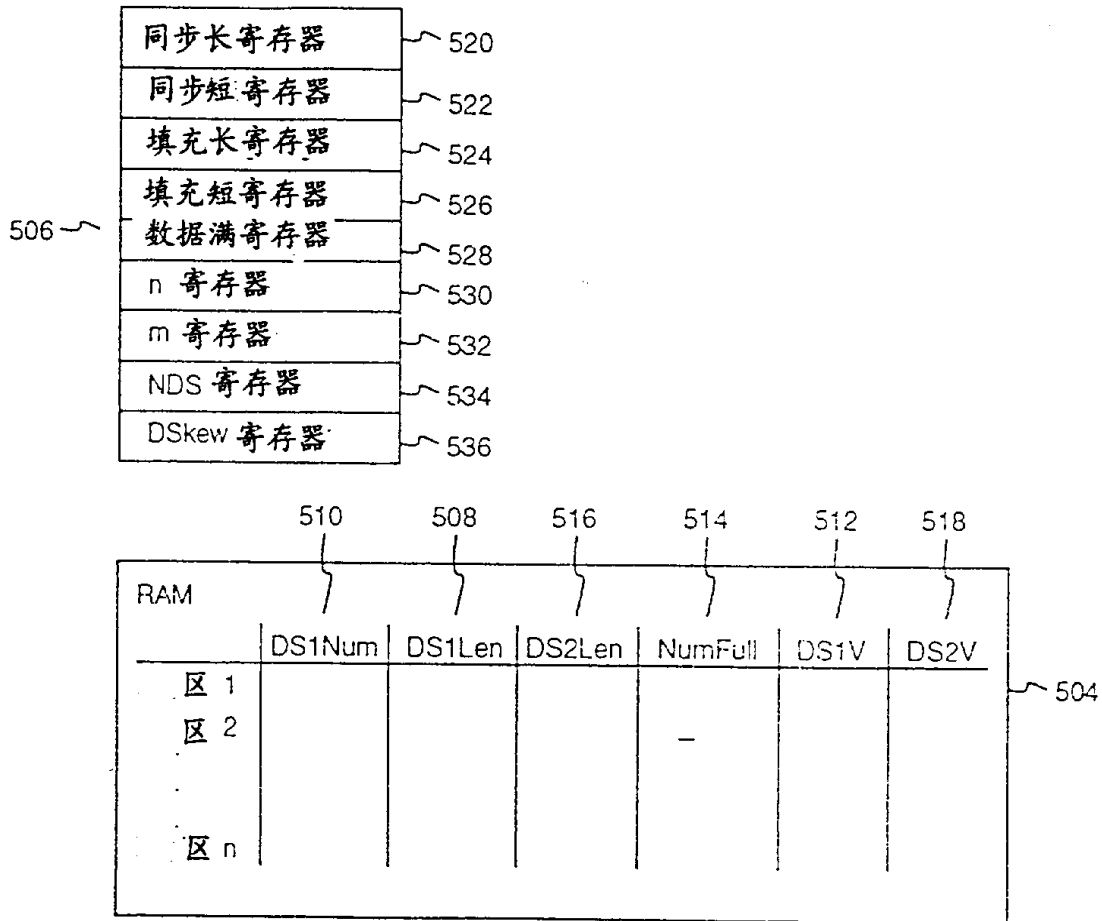


图 5 扇区计算存储器的组成

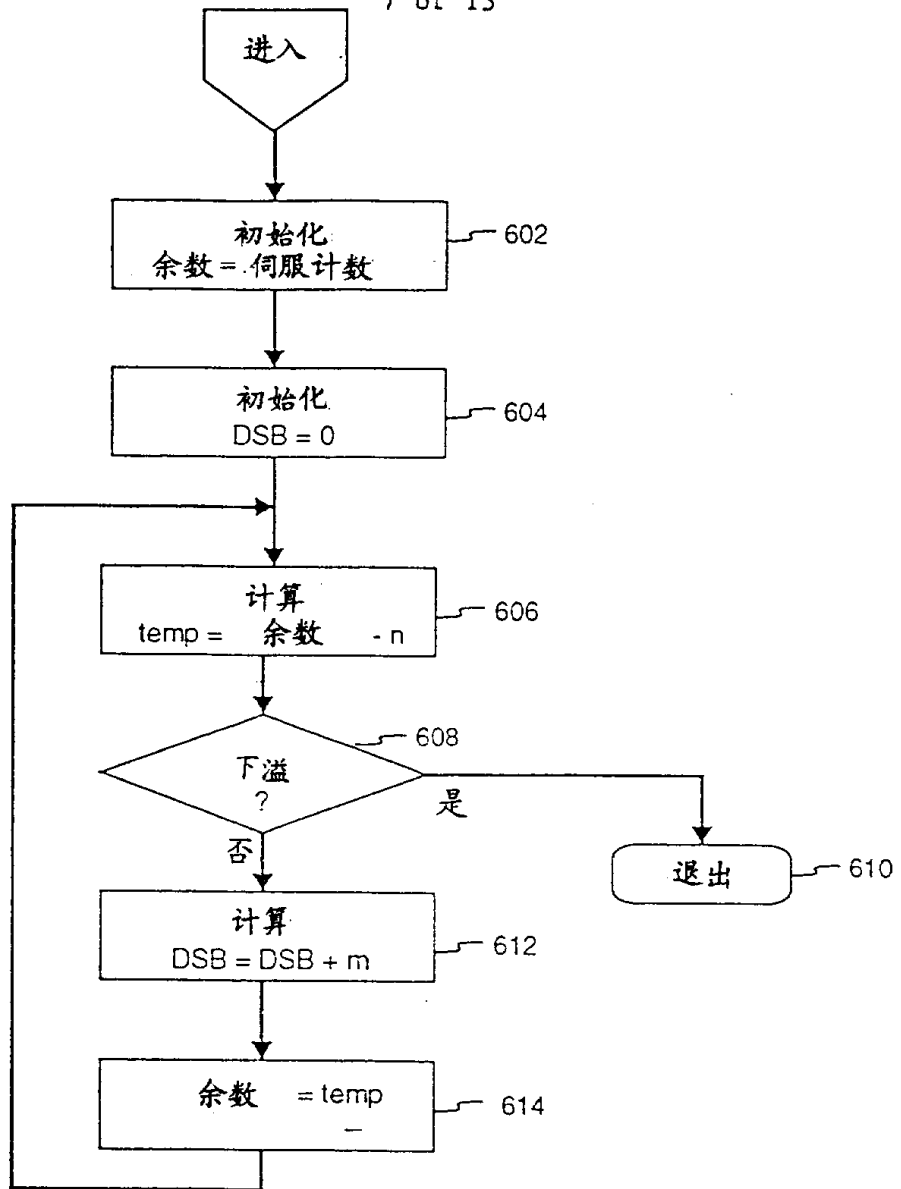


图 6 伺服模数计算流程

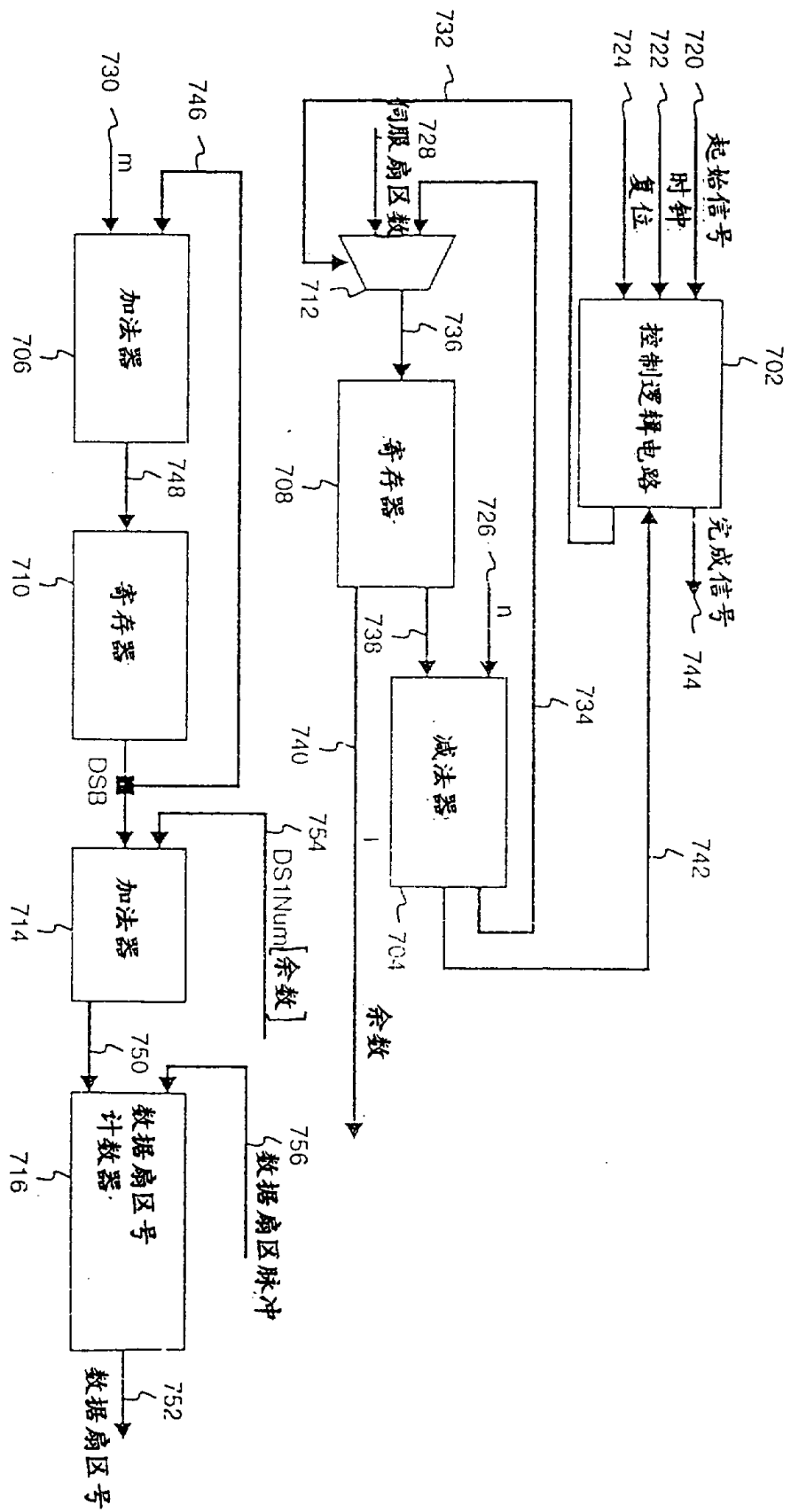
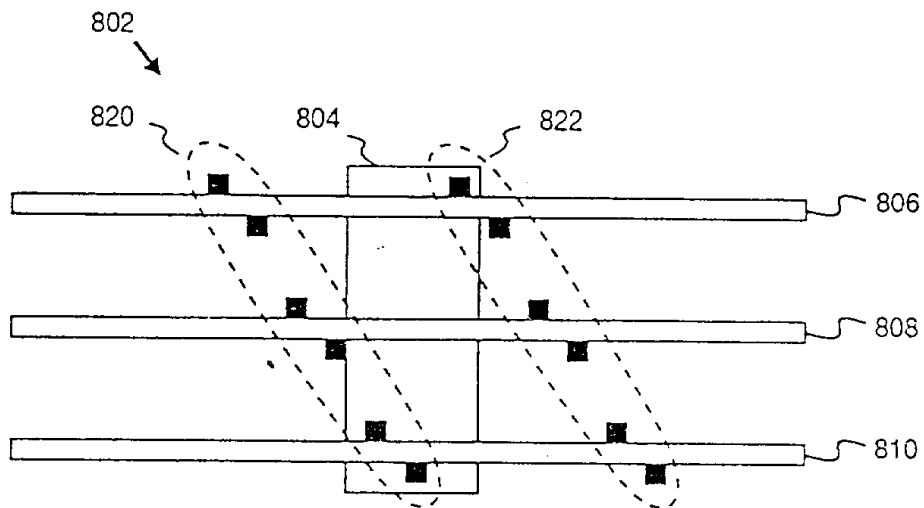


图7 伺服模数计数器



磁头位移量	偏移量	伺服时序调整量
-5	+1	+1/6
-4	+1	+2/6
-3	+1	+3/6
-2	+1	+4/6
-1	+1	+5/6
+1	0	+1/6
+2	0	+2/6
+3	0	+3/6
+4	0	+4/6
+5	0	+5/6

图 8. 带有交错伺服记录的磁盘驱动器

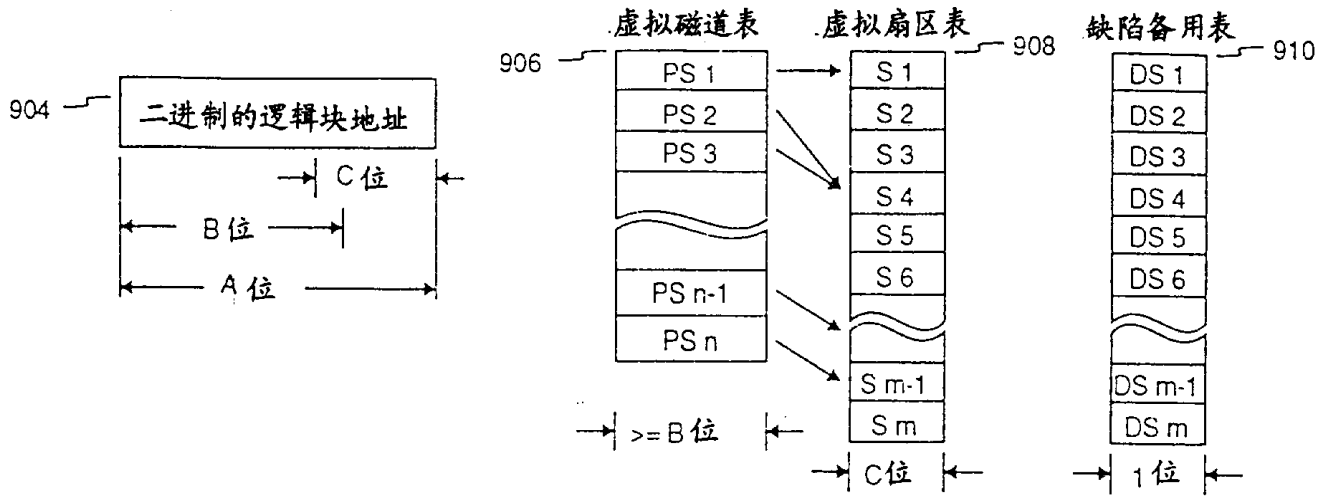


图 9. 逻辑块地址到物理块地址的映射结构

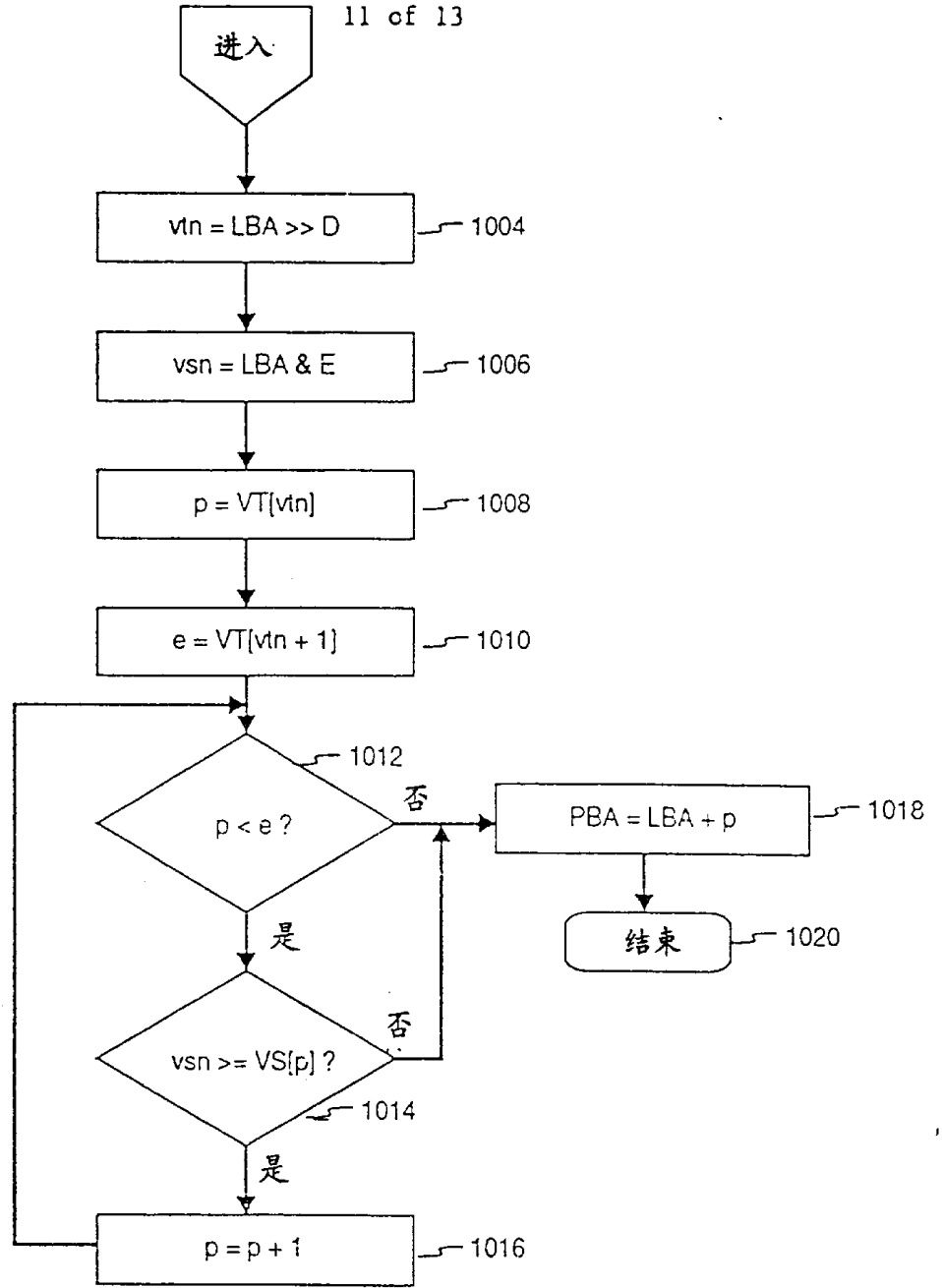


图 10 逻辑块地址至物理块地址转换计算流程

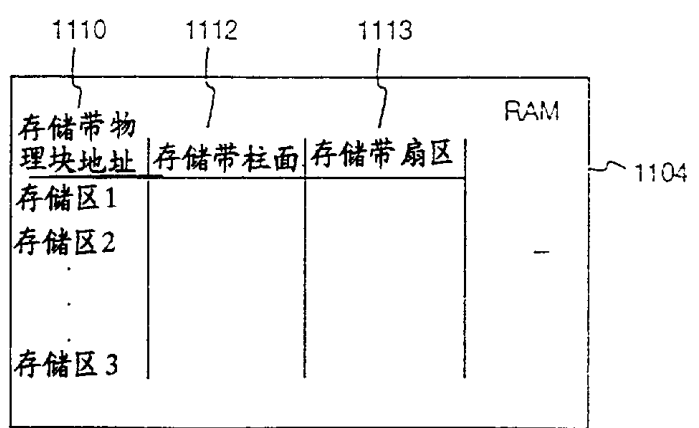


图 11 存储带转换存储器的组成

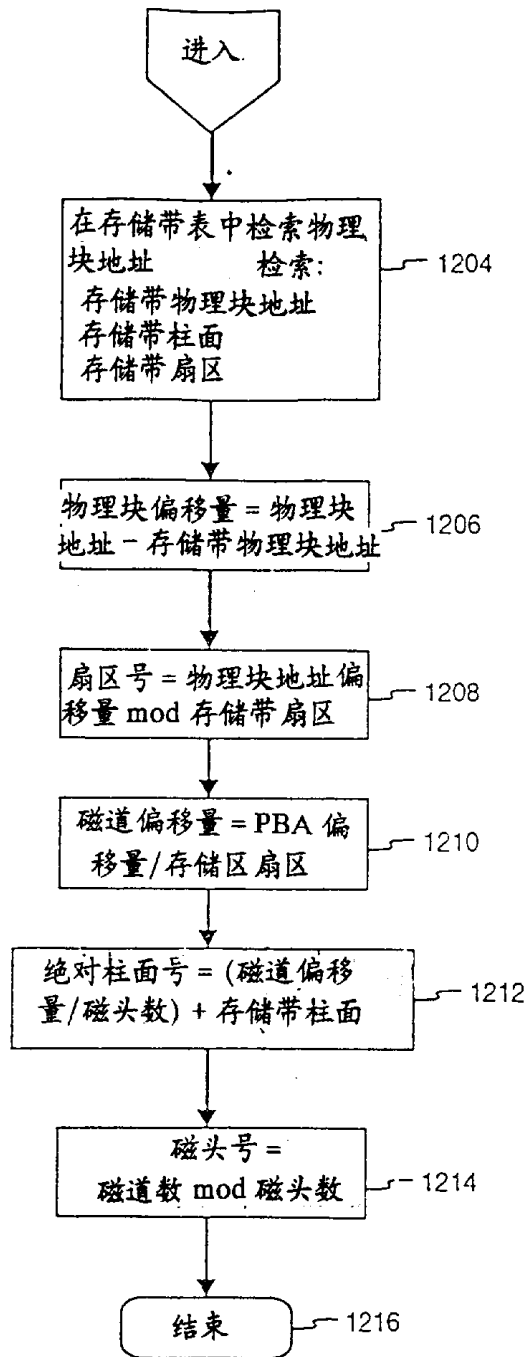


图 12: PBA 到存储带、柱面、磁头及扇区的计算