[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H03M 7/30 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200880016010.7

[43] 公开日 2010 年 3 月 31 日

[11] 公开号 CN 101689863A

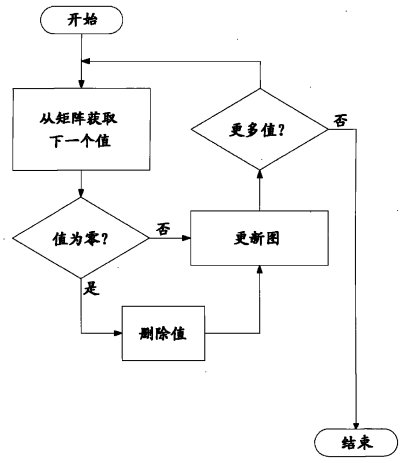
[22] 申请日 2008.3.14
[21] 申请号 200880016010.7
[30] 优先权
[32] 2007. 3.15 [33] GB [31] 0704976.0
[32] 2007. 4.11 [33] US [31] 60/911,273
[86] 国际申请 PCT/EP2008/053133 2008.3.14
[87] 国际公布 WO2008/110633 英 2008.9.18
[85] 进入国家阶段日期 2009.11.13
[71] 申请人 线性代数技术有限公司
地址 爱尔兰都柏林
[72] 发明人 大卫·莫洛尼
[74] 专利代理机构 北京集佳知识产权代理有限公司
代理人 潘士霖 李春晖
权利要求书 6 页 说明书 13 页 附图 9 页

[54] 发明名称

用于压缩数据的电路和利用该电路的处理器

[57] 摘要

本申请解决计算系统设计中使存储器访问成本最小这一根本问题。 这是对计算机系统设计的一个根本限制，因为无论存储器技术或者连接到处理器的方式如何，都存在对在给定时间内可以在处理器与存储器之间传送多少数据的最大限制，这就是可用存储器带宽，并且可用存储器带宽对计算能力的限制通常称为“存储墙”。 提供的解决方案创建要被压缩的数据结构的图，该图代表非平凡数据值(例如非零值)在结构中的位置并且从结构删除平凡数据值以提供压缩结构。



1. 一种压缩电路，包括:

a）数据存储器，用于存储各个数据值的结构，

b）图存储器，用于存储图，所述图代表非零值在所述结构内的位置，

c）数据输出，其中所述电路被配置成使用所述图从所述数据存储器获取非零数据并且在所述数据输出上结合代表所述图的数据来提供所述获取的数据作为压缩结构。

2. 根据权利要求 1 所述的电路，其中所述数据存储器包括多个寄存器。

3. 根据权利要求 1 所述的电路，其中所述数据存储器包括寄存器堆。

4. 根据权利要求 1 所述的电路，其中所述数据存储器包括寄存器。

5. 根据权利要求 4 所述的电路，其中所述位置被存储为位图。

6. 根据权利要求 5 所述的电路，其中所述位图中的每个位与所述存储的结构中的单独的数据值相对应。

7. 根据权利要求 1 所述的电路，还包括多个比较器，每个比较器标识数据值是否为非零，每个比较器的输出作为输入提供给所述图存储器。

8. 根据权利要求 7 所述的电路，其中所述比较器输入由所述数据存储器的读端口提供。

9. 根据权利要求 7 所述的电路，其中所述比较器输入由所述数据存储器的写端口提供。

10. 根据权利要求 7 所述的电路，其中所述比较器输入由处理器加载/存储端口提供。

11. 根据权利要求 2 所述的电路，其中所述数据输出包括数据总线，并且所述电路被配置成将所述压缩结构从所述存储器依次输出到所述数据总线上。

12. 根据权利要求 2 所述的电路，还包括用于根据所述图来计算非零值数目的至少一个加法器。

13. 根据权利要求 2 所述的电路，还包括用于依次实现将非零数据从所述存储器写入到所述数据输出的逻辑。

14. 根据权利要求 13 所述的电路，其中所述逻辑包括加法器的布置。

15. 根据权利要求 14 所述的电路，其中所述布置中的每个后续加法器以所述布置中的先前加法器的输出作为输入。

16. 根据权利要求 2 所述的电路，其中每个加法器与所述结构中的关联数据值相对应，并且每个加法器接受与所述关联数据值相对应的来自所述图的输入。

17. 根据权利要求 14 所述的电路，还包括整数比较器的树，每个整数比较器用于比较两个整数输入，每个比较器的第一输入是来自所述加法器树中的对应加法器的输出。

18. 根据权利要求 17 所述的电路，其中每个比较器的第二输入是定序信号。

19. 根据权利要求 17 所述的电路，还包括用于组合来自所述图的值与各个比较器输出的组合器以保证将非平凡数据写入到所述数据输出的正确顺序。

20. 根据权利要求 1 所述的电路，还包括用于控制所述电路的操作的控制器。

21. 根据权利要求 1 所述的电路，其中所述数据值是单精度浮点数。

22. 根据权利要求 1 所述的电路，其中所述数据值是双精度浮点数。

23. 根据权利要求 1 所述的电路，其中所述数据值是扩展精度浮点数。

24. 根据权利要求 1 所述的电路，其中所述数据值是 128 位精度浮点数。

25. 根据权利要求 1 所述的电路，其中所述数据值是整数。

26. 根据权利要求 1 所述的电路，其中所述电路适用于也提供所述图的内容作为输出。

27. 根据权利要求 1 所述的电路，其中在集成电路上提供所述电路。

28. 一种处理器，包括至少一个根据权利要求 1 至 27 中的任一权利要求所述的电路。

29. 根据权利要求 28 所述的处理器，其中有所述电路的多个实例。

30. 一种用于根据压缩结构来提供解压缩结构的解压缩电路，所述电路包括：

a）输入，用于接收各个非平凡数据值的压缩结构，

b）图寄存器，用于接收标识所述非平凡数据值在所述解压缩结构内的位置的图，

c）存储器，用于存储所述解压缩结构，其中所述电路被配置成根据所述图寄存器的内容用各个输入的非平凡数据值填充所述存储器。

31. 根据权利要求 30 所述的电路，其中所述存储器包括多个寄存器。

32. 根据权利要求 30 所述的电路，其中所述存储器包括寄存器堆。

33. 根据权利要求 30 所述的电路，其中所述位置存储为位图。

34. 根据权利要求 33 所述的电路，其中所述位图中的每个位与所述解压缩结构中的单独的数据值相对应。

35. 根据权利要求 30 所述的电路，其中所述数据输入包括数据总线，并且所述电路被配置成将所述压缩结构从所述数据总线依次输入到所述存储器中。

36. 根据权利要求 30 所述的电路，还包括用于根据所述图来计算非平凡值的数目的至少一个加法器。

37. 根据权利要求 30 所述的电路，还包括用于依次实现将非平凡数据从所述数据输入写入到存储器的逻辑。

38. 根据权利要求 37 所述的电路，其中所述逻辑包括加法器的布置。

39. 根据权利要求 38 所述的电路，其中所述布置中的每个后续加法器以所述布置中的先前加法器的输出作为输入。

40. 根据权利要求 39 所述的电路，其中每个加法器与所述未压缩结构中的关联数据值相对应，并且每个加法器接受与所述关联数据值相对应的来自所述图的输入。

41. 根据权利要求 40 所述的电路，还包括整数比较器的布置，每个整数比较器用于比较两个整数输入，每个比较器的第一输入是来自所述加法器的布置中的对应加法器的输出。

42. 根据权利要求 41 所述的电路，其中每个比较器的第二输入是定序信号。

43. 根据权利要求 41 所述的电路，还包括用于组合来自所述图的值与各个比较器输出的组合器以保证将非平凡数据写入到所述数据输出的正确顺序。

44. 根据权利要求 41 所述的电路，还包括用于控制所述电路的操作的控制器。

45. 根据权利要求 30 所述的电路，其中所述数据值是单精度浮点数。

46. 根据权利要求 30 所述的电路，其中所述数据值是双精度浮点数。

47. 根据权利要求 30 所述的电路，其中所述数据值是整数。

48. 根据权利要求 30 所述的电路，还包括图输入，其中所述电路适用于将图从所述图输入加载到所述图寄存器中。

49. 根据权利要求 30 所述的电路，其中在集成电路上提供所述电路。

50. 根据权利要求 30 至 49 中的任一权利要求所述的电路，其中平凡数据值是零数据值，而所述非平凡数据值为非零数据值。

51. 一种处理器，包括至少一个根据权利要求 30 至 50 中的任一权利要求所述的电路。

52. 根据权利要求 51 所述的处理器，其中有所述电路的多个实例。

53. 一种处理器芯片，包括响应于用以存储数据结构的指令的压缩电路，所述压缩电路适用于从所述结构去除平凡值以提供用于存储的压缩格式。

54. 根据权利要求 53 所述的处理器芯片，其中所述压缩电路适用于提供标识平凡值在所述结构中的位置的图。

55. 根据权利要求 53 所述的处理器，还包括响应于用以加载压缩格式数据的指令的解压缩电路，所述解压缩电路适用于在加载所述压缩数据时将平凡值重新填充到所述压缩数据中。

56. 根据权利要求 55 所述的处理器，其中所述解压缩电路利用图以重新填充平凡值。

57. 根据权利要求 53 至 56 中的任一权利要求所述的处理器，其中平凡数据值是零数据值，而所述非平凡数据值为非零数据值。

58. 根据权利要求 1 所述的电路，其中所述电路被配置成在所述数据输出上并行提供多个所述获取的单独的数据。

59. 根据权利要求 58 所述的电路，其中所述各个数据值的长度是 x 位，并且所述数据输出包括 nx 位数据总线，其中 n 是大于一的整数，并且 n 个数据值一次放置于所述数据总线上。

60. 根据权利要求 30 所述的电路，其中所述电路被配置成接收各个非平凡数据值的多个所述压缩结构。

61. 根据权利要求 60 所述的电路，其中所述各个数据值的长度是 x 位，并且所述数据输入包括 nx 位数据总线，其中 n 是大于一的整数，并且从所述数据总线一次获取 n 个数据值。

62. 一种压缩数据值结构的方法，所述方法包括以下步骤：创建标识零值在所述结构内的位置的图；并且从所述结构去除标识的平凡条目值以提供仅包括所述非零值和所述图的压缩结构。

63. 根据权利要求 62 所述的方法，其中所述数据值是浮点数。

64. 根据权利要求 62 所述的方法，其中所述数据值是单精度浮点数或者双精度浮点数。

65. 根据权利要求 62 所述的方法，其中所述数据值是扩展精度浮点数或者 128 位精度浮点数。

66. 根据权利要求 62 所述的方法，其中所述数据值是整数。

67. 根据权利要求 62 至 66 中的任一权利要求所述的方法，其中所述图包括位图，其中所述位图的每个位代表单独的数据值。

68. 根据权利要求 62 至 67 中的任一权利要求所述的方法，其中所述标识所述位置的步骤包括比较每个数据值以确定它是否为非零值。

69. 根据权利要求 68 所述的方法，其中对每个比较的输出进行求和以提供对非零值数目的计数。

70. 根据权利要求 68 所述的方法，其中所述计数用来确定所述压缩结构的大小。

71. 根据权利要求 68 所述的方法，其中所述计数用来确定要在所述压缩结构中提供的条目的数目。

72. 根据权利要求 68 所述的方法，其中每个比较的输出用来实现将数据值写入到所述压缩结构。

73. 根据权利要求 62 至 72 中的任一权利要求所述的方法，其中所述结构是矩阵，并且所述图标识列和行的数目。

74. 根据权利要求 62 所述的方法，其中所述结构包括按照行-列布置来布置的矩阵。

75. 一种压缩数据结构，包括多个非零数据值和图，所述图代表零数据值相对于多个非平凡数据值在所述结构的非压缩形式中的位置。

76. 根据权利要求 75 所述的压缩数据结构，其中所述图包括位图，其中每个单独的数据值由单独的位代表。

77. 一种对压缩数据结构进行解压缩的方法，所述压缩结构包括多个非零数据值和图，所述图代表所述非零值在未压缩结构中的位置，所述方法包括以下步骤:

提供未填充的未压缩结构，

接收所述非零值，并且

根据所述非零值的在所述图中代表的位置在所述未填充的结构内填充所述非零值以提供填充的解压缩数据结构。

78. 根据权利要求 77 所述的方法，其中所述未填充的矩阵中的值在填充所述非零值之前被初始化成零。

79. 根据权利要求 77 所述的方法，其中所述填充的结构中在所述图中标识为零值的位置被设置成零。

80. 根据权利要求 77 所述的方法，其中所述数据值是浮点数。

81. 根据权利要求 77 所述的方法，其中所述数据值是单精度浮点数。

82. 根据权利要求 77 所述的方法，其中所述数据值是双精度浮点数。

83. 根据权利要求 77 所述的方法，其中所述数据值是整数。

84. 根据权利要求 77 至 83 中的任一权利要求所述的方法，其中所述图包括位图，其中所述位图的每个位代表单独的数据值。

85. 根据权利要求 84 所述的方法，其中对所述位图的各个位进行求和以提供对所述压缩结构中的非零值数目的计数。

86. 根据权利要求 85 所述的方法，其中所述计数用来确定要被读入到所述未填充的结构中的数据值的数量。

87. 根据权利要求 77 至 86 中的任一权利要求所述的方法，其中所述图用来实现将数据值写入到所述未压缩结构。

99. 根据权利要求 77 至 87 中的任一权利要求所述的方法，其中所述结构包括具有行-列配置的矩阵。

用于压缩数据的电路和利用该电路的处理器

技术领域

本申请涉及数据压缩方法并且具体地涉及结构的压缩。

背景技术

压缩方法和算法在用于减少要存储于存储器中的数据量的领域中众所周知。具体而言，已知用于不同数据类型的不同算法，例如用于图像的JPEG。

本申请涉及矩阵结构的压缩。

先前尝试的一种方法（Moloney 和 Geraghty WO2006120664）涉及到通过将未结构化矩阵转换成结构化矩阵来压缩矩阵结构。该压缩方法有效地消除在矩阵的主对角线以上和以下的数据重复。

在 US6,591,019 中提出了用于处理矩阵结构的另一方法，其中将数据值矩阵压缩成包括位图表、符号图表和数据图表的结构。位图表包括 2 位条目串，每个 2 位条目与未压缩矩阵结构中的条目相对应。位图中的每个两位条目标识非压缩矩阵中的对应值是零还是一或者是以缩放（scaling）形式存储于数据图中还是以未压缩形式存储于数据图中。符号图表标识非压缩结构的值的符号。这种方法的弊端在于它不是无损的，因为信息在缩放值时受到损失。该方法仅适用于在软件中实施。

虽然上述每种方法就需要更少存储器以在存储器中存储结构而言提供了相对于现有技术的改进，但是这些方法当在处理器中解压缩结构时仍然造成大量计算成本。此外，上述两种方法还由于它们在压缩和解压缩方面的相对复杂性而受到处理速度的困扰，压缩和解压缩当在处理器上运行时导致装置相对庞大并且性能缓慢。

本申请解决在计算系统设计中使实施成本最小而使该实施的压缩\解压缩速度最大之时使存储器访问成本最小的问题。存储器访问（存储器带宽）的成本是对计算机系统设计的根本限制，因为无论存储器技术或者连接到处理器的方式如何，都存在关于在给定时间内可以在处理器与存储器

之间传送多少数据的最大限制，这就是可用存储器带宽，并且可用存储器带宽对计算能力的限制常称为“存储墙”。

本发明寻求通过以压缩格式存储数据并且提供适合于在许多应用中使用的块结构化数据的压缩和解压缩手段来增加有效存储器带宽并且使“存储墙”对计算的限制最小，其中所述应用比如对必须存储于存储器中的大型数据集进行操作的计算机图形、刚性体动力学、有限元分析以及其它科学和工程应用。

发明内容

本申请通过将压缩\解压缩电路并入到处理器的硬件中、使得数据传送和解压缩没有占用处理器资源并且在保持实施成本低和处理速度快的同时没有大量延迟来解决存储墙的问题。提供一种允许相对简易的压缩\解压缩硬件的有利的压缩方法。

第一实施例提供一种电路，包括：存储器，用于存储各个数据值的结构；图存储器，用于存储非平凡（非零）值在结构内的位置；数据输出，其中该电路被配置成使用图从存储器获取非平凡（非零）数据并且在数据输出上提供获取的数据作为压缩结构。存储器适当地包括一个寄存器、多个寄存器和/或寄存器堆（register file）。位置可以存储为位图，位图中的每个位可以对应于存储的结构中的单独的数据值。可以提供多个比较器，每个比较器标识数据值是否为非平凡，每个比较器的输出作为输入提供给图。比较器输入可以由存储器的读端口、存储器的写端口和/或处理器加载/存储端口（load/store port）来提供。

数据输出可以包括数据总线，并且该电路被配置成将压缩结构从存储器依次输出到数据总线上。该电路还可以包括用于根据图来计算非零值数目的至少一个加法器。该电路还可以包括用于依次实现将非零数据从存储器写入到数据输出的逻辑。该逻辑可以包括加法器的布置，其中该布置中的每个后续加法器以该布置中的先前加法器的输出作为输入。每个加法器可以与结构中的关联数据值相对应，并且每个加法器接受来自与关联数据值相对应的图的输入。该电路还可以包括整数比较器的树，每个整数比较器用于比较两个整数输入，每个比较器的第一输入是来自加法器树中的对应加法器的输出。每个比较器的第二输入是定序信号。该电路还可以包括用于组合来自图的值与各个比较器输出的组合器以保证将非平凡数据写

入到数据输出的正确顺序。该电路还可以包括用于控制电路的操作的控制器。该电路可以也适用于提供图的内容作为输出。适当地，可以在集成电路上提供该电路。可以提供包括一个或者多个该电路的处理器。

在又一实施例中，提供一种用于从压缩结构提取解压缩结构的电路。该电路包括：输入，用于接受各个非平凡数据值的压缩结构；图寄存器，用于接收标识非平凡数据值在解压缩结构内的位置的图；存储器，用于存储解压缩结构，其中该电路被配置成根据图寄存器的内容用各个输入的非平凡数据值填充存储器。存储器可以包括多个寄存器或者寄存器堆。位置可以存储为位图，该位图中的每个位与解压缩结构中的单独的数据值相对应。

数据输入可以包括数据总线，并且在该情况下该电路被配置成将压缩结构从数据总线依次输入到存储器中。该电路还可以包括用于根据图来计算非平凡值数目的至少一个加法器。该电路还可以包括用于依次实现将非平凡数据从数据输入写入到存储器的逻辑、适当地为加法器的布置。该布置中的每个后续加法器可以将该布置中的先前加法器的输出作为输入。每个加法器可以与未压缩结构中的关联数据值相对应，并且每个加法器接受与关联数据值相对应的来自图的输入。该电路还可以包括整数比较器的布置，每个整数比较器用于比较两个整数输入，每个比较器的第一输入是来自加法器的布置中的对应加法器的输出。每个比较器的第二输入可以是定序信号。该电路还可以包括用于组合来自图的值与各个比较器输出的组合器以保证将非平凡数据写入到数据输出的正确顺序。该电路还可以包括控制器，用于控制电路的操作。该电路还可以包括图输入，其中该电路适用于将图从图输入加载到图寄存器中。适当地，可以在集成电路上提供该电路。适当地，平凡数据值是零数据值，而非平凡数据值为非零数据值。本申请扩展到并入至少一个这些电路的处理器。

又一实施例提供一种处理器芯片，该芯片包括响应于用以存储数据结构的指令的压缩电路，该压缩电路适用于从结构去除平凡值以提供用于存储的压缩格式。该压缩电路可以适用于提供标识平凡值在结构中的位置的图。该处理器还可以包括响应于用以加载压缩格式数据的指令的解压缩电路，该解压缩电路适用于在加载压缩数据时将平凡值重新填充到压缩数据中。该解压缩电路可以利用图以重新填充平凡值。适当地，平凡数据值是零数据值，而非平凡数据值为非零数据值。

因而，另一实施例提供一种压缩数据值结构的无损方法，该方法包括

以下步骤：创建标识平凡条目值在结构内的位置的单个图；并且从结构去除所标识的平凡条目值以提供仅根据单个图中的信息即可解压缩的压缩结构。数据值可以是浮点数（单精度或者双精度）、扩展精度浮点数、128位精度浮点数或者整数。

适当地，该图包括位图，其中该位图的每个位代表单独的数据值。标识位置的步骤可以包括比较每个数据值以确定它是否为非平凡条目值。可以对每个比较的输出进行求和以提供对非平凡条目值数目的计数。该计数可以用来确定压缩结构的大小和/或将要在压缩结构中提供的条目的数目。每个比较的输出可以用来实现将数据值写入到压缩结构。该结构可以是矩阵，而该图可以标识列和行的数目。在一个有利布置中，平凡条目值是零值，而非平凡值为非零。

又一实施例提供一种压缩数据结构，该结构包括多个非平凡数据值和图，该图代表平凡数据值相对于多个非平凡数据值的位置。在一个有利布置中，平凡条目值是零值，而非平凡值为非零。该图可以包括位图，其中每个单独的数据值由单独的位代表。

又一实施例提供一种对压缩数据结构进行解压缩的方法，该压缩结构包括多个非平凡数据值和图，该图代表非平凡值在未压缩结构中的位置。该方法包括以下步骤：提供未填充的未压缩结构；获取非平凡值；并且根据非平凡值的在图中代表的位置在未填充的结构内填充非平凡值以提供填充的解压缩数据结构。在一个有利实施例中，平凡条目值是零值，而非平凡值为非零。在这种情况下，未填充的矩阵中的值可以在填充非零值之前初始化成零。可替换地，填充的结构中在该图中标识为零值的位置可以设置成零。该图可以包括位图，其中该位图的每个位代表单独的数据值。可以对位图的各个位进行求和以提供对压缩结构中的非零值数目的计数。该计数可以用来确定将要读入到未填充的结构中的数据值的数量。适当地，该图用来实现将数据值写入到未压缩结构。该结构可以包括具有行-列配置的矩阵。

这些和其它实施例、特征及优点将从以下示例性描述中变得清楚。

附图说明

现在将参照附图描述本发明，在附图中：

图 1 是根据本申请的压缩方法的示例性流程图；

图 2 是未压缩结构及其作为图 1 的方法的结果所得的压缩结构;

图 3 是根据本申请的解压缩方法的示例性流程图;

图 4 是根据本申请的示例性处理器;

图 5 是适合于包括在图 4 的处理器中的示例性解压缩电路;

图 6 是说明图 5 的操作的示例性时序图;

图 7 是适合于包括在图 4 的处理器中的示例性解压缩电路;

图 8 是说明图 7 的操作的示例性时序图; 并且

图 9 是示出可以如何组合图 5 和图 7 的电路的示例性结构。

具体实施方式

本申请采用一种利用在大型浮点矩阵数据集(比如在 3D 计算机图形、游戏物理学(刚性体动力学)、有限元分析(FEA)和搜索引擎中使用的数据集)中包含的零数据值的新方法。然而,本申请适用于其它数据结构而不仅限于基于矩阵的结构。

发明人已经认识到对于许多矩阵而言大量条目是零填充,但是这些零占据浮点表示的 32 位或者 64 位,这些零必须从片上或者片外存储器来取回(fetch)并且可能使处理器忙于对从存储器或者寄存器取回的零进行平凡(trivial)操作。

本申请提供在已经去除零数据值的压缩结构中存储按照数据值的行和列布置的矩阵结构。为压缩结构提供关联图,该图标识零值和非零值在矩阵结构中的位置并且允许对矩阵结构的无损(loss-less)重建。将理解这样的图可以被视为标识平凡值的位置或者标识非平凡值的位置,因为不言而喻如果值没有被标识为平凡值,则它为非平凡值并且反之亦然。这种方法提供相对于现有技术的多个显著优点。首先,该方法为无损,并且其次该方法可以易于以硬件实施,因此减少了与在加载到存储器中时对压缩结构进行解压缩通常关联的计算负荷。类似地,可以用压缩形式存储数据而不造成附加的处理资源。

这种图可以与压缩结构一起或者作为索引结构(其中需要随机访问复杂数据结构)的部分来存储于存储器中。

本技术的优点在于它在假设32位的位图用来代表多达32个数据结构

条目的情况下，将针对零值数从存储器的传送要求对于单精度值从 32 位减少至 1 位或者对于双精度值从 64 位减少至 1 位。显然可以调节位图大小以提供具有多于或者少于 32 个条目的数据结构而不失一般性。对于整数值也可以获得相同优点。

现在将参照图 1 的示例性顺序流程图和图 2 的包含十六个 32 位单精度条目的示例性 4×4 矩阵来描述操作的模式。

该处理通过从要被压缩的矩阵结构获取（retrieve）第一条目来开始。比较第一条目以确定它是零值还是非零值。根据比较结果，在图的第一位置产生条目以将第一个值标识为零值或者非零值。适当地，该图是其中矩阵结构中的每个值具有该位图中的关联位的位图。在第一条目是零值的情况下，丢弃、删除该条目或者标记该条目用于以后处理或者删除（例如在向压缩的储存器传送矩阵数据期间），并且在用于矩阵的图的第一位置产生条目以表明存在零值、例如通过将第一位设置成零。在矩阵中的第一个值为非零值的情况下，例如图 2 中的第一条目是 1.0，将图中的第一条目设置成 1 来表明非零值。同时将第一条目存储为压缩结构中的第一条目。

然后针对矩阵中的每个其余条目重复该处理。在所示示例性矩阵中，逐行扫描条目，尽管将理解也可以采用逐列扫描或者可以采用多个比较器来在单个操作中而不是依次使用单个比较器来生成位图的整行/列。

因此对于具有 1.0、2.0、3.0 和 4.0 这些值的第一行，对应的图是 1111，因为没有一个条目是零值，而在第二行中，图是 0100，因为仅第二个值为非零值（6.0）。类似地，对于第三行，只有图中的最后的条目是一，因为该行中的前三个值是零。

压缩处理的结果是先前在存储器中存储为 16 个条目（对于单点精度而言每个条目为 32 位）的矩阵可以由 9 个条目（每个条目也为 32 位）的压缩矩阵和代表这些非零值和对应的零值在非压缩矩阵中的位置的 16 位图取代。由于每个存储器位置是 32 位，所以需要 32 位而不是 16 位用于存储图，其中图与矩阵一起存储。然而，16×32 位未压缩矩阵格式（总共 512 位）已经由 9×32 位压缩矩阵和 1×32 位图（总共 320 位）取代，这代表压缩(512-320)/512=37.5%。

只要每个稠密矩阵（包括零填充）的 1 个或者多个条目是零就实现压缩(在采用 32 位位图的情况下)；然而对于非零条目每个条目有 1 位损失，这可能导致更多存储器用来存储无零填充的稠密数据集。在实践中，包括

3D 计算机图形、游戏物理学（刚性体动力学）、有限元分析（FEA）和搜索引擎（例如 Google）的令人感兴趣的数据集是稀疏的并且包含大量零填充。

尽管按照多行和多列描述本申请，但是它也适用于压缩包括单行或者单列的结构。

将理解在一些情况下可能有必要知道矩阵的列-行结构、即行和列的数目，以便重构矩阵。然而在大量应用中，这在很大程度上无关紧要，因为从存储器获取内容并且以标量形式（即如图 2 的单列中所示）在寄存器中存储内容，并且这是将数据视为矩阵的软件\处理器。此外，假如在压缩与解压缩方式之间存在一致性，行-列结构可以隐含地存在。在需要知道列-行结构的情况下，这可以包括在图内\附加到图。

如图 3 中所示将压缩结构扩展成未压缩结构的处理是压缩方法的逆过程。该方法通过提供空的未压缩结构并且加载图来开始。然后比较图中的第一位以确定它是零值还是为非零值。当第一位是一时，将第一数据值从存储器加载到未压缩结构的第一条目中。类似地，当第一位是零时，将零值插入到未压缩结构中的第一条目中。在未压缩结构中的所有值已经被初始化成零的情况下，将无需具体的插入零值的步骤。重复该处理直至已经达到图的末尾或者直至已经加载所有数据值。可以简单地通过对图中的各个非零位（对应于压缩非零矩阵/矢量条目）进行计数来计算要被加载的数据值的数目。

在对图 2 的压缩结构进行解压缩的情况下，由于图中的前四个值是 1，所以前四个数据值将从压缩结构加载到未压缩结构的前四个条目中。由于图中的第五位是零，所以未压缩结构中的第五条目将被填充（populate）零值，等等。

即使当将压缩结构扩展成未压缩结构时，仍然可以保持用于压缩结构的图并且随后显著有利地利用该图。具体而言，如在申请人的共同未决申请中所描述的，作为乘法、加法或者其它算术运算的结果，可以有利地使用图来控制处理器内的功能单元。

为求效率，当矩阵结构的大小相对大时，它可以划分成多个子矩阵，每个子矩阵代表矩阵结构的一部分。在这种布置中，可以用上述方式单独地压缩每个子矩阵。

虽然可以用软件实施上述方法，但是当在如图 4 中所示从处理器和向

处理器移动数据时在硬件中解压缩和压缩数据时，可以获得在处理速度方面的显著优点。将理解所选择的压缩结构与包括缩放和其它操控的现有技术方法相比尤其适合于在硬件中实施。在硬件布置中，处理器可以适用于具有用于存储和/或获取块形式的数据、例如存储\获取矩阵的具体指令。在这样的布置中，压缩和解压缩处理\硬件可以对于中央处理器芯是透明的，尽管如上文所说明的那样可以显著有利地在处理器本身中利用图并且类似地如将要讨论的那样可以有利地在处理器硬件内\结合处理器硬件来生成图。将理解在任何情况下，可以有利地以对于程序员\操作软件基本上透明的方式进行压缩\解压缩的操作。

还将理解，虽然已经参照将数据从片上或者片外存储器加载和存储到处理器描述了本申请，但是将理解各种其它应用是可能的，这些应用包括例如在协同处理器之间共享数据。

现在将描述在将压缩数据结构从存储器加载到处理器时提供对该压缩数据结构的解压缩扩展的示例性硬件实施。出于示例目的，使用下表 1 中所示相对小的 3×3（9 个条目）压缩矩阵，其中 A、B、C、D 和 E 代表非零值，而 101010101 是用于 3×3 矩阵中的非零值的图。在实践中，所提出的技术可以扩展到任何任意大小的 N×M 矩阵和用于作为用于解压缩数据的目的地的、具有大于或者等于 N×M 个条目的寄存器堆。

A
B
C
D
E
101010101

表 1

如图 5 中所示的示例性解压缩逻辑包括压缩位图寄存器、写地址计算器、写地址比较器、组合器和目的地片上存储器。在示例性布置中，这些单元在控制器的控制之下，例如地址生成单元（AGU）或者直接存储器访问（DMA）控制器。现在将参照图 6 的示例性时序图更具体地说明这些单元的构造和操作。

压缩位图寄存器是大小足以存储用于压缩矩阵结构的图的寄存器，在

实践中这对应于用于要支持的N×M压缩矩阵中的每个条目的1位条目。在示例性布置中，压缩位图寄存器包括9位（MCB0-MCB8），每个位与未压缩结构中的位置相对应。在扩展处理开始时，控制器将图从片上或者片外存储器加载到压缩位图寄存器中。将理解图5中的位图寄存器中所示的值完全是示例性的值。类似地，在为压缩矩阵数据生成读地址的控制器的控制之下通过数据_输入总线从存储器串行发送值数据。可替换地，可以按照非零条目的组从存储器发送数据，例如尽管以增加布线为代价,可以通过64位总线发送2×32位的值而不失一般性。控制器的动作由解压缩逻辑和压缩位图寄存器控制，并且允许时钟周期使用写地址计算器来生成从基地址开始将要取回的字节数目（字节_取回）。

写地址计算器包括2进制补码整数加法器树以针对片上存储器中的每个条目（寄存器-文件）计算写地址（WA）。树中的每个加法器具有两个输入。第一输入是来自位图寄存器的对应值、即位图寄存器中的第一条目（MCB0）是向第一加法器的输入。类似地，位图寄存器中的最后条目（MCB8）是向树中的最后加法器的输入。向每个加法器的第二输入包括来自树中的前一加法器（WA0）的输出，因此第一加法器（WA0）的输出是向第二加法器的输入，并且类似地，第八加法器（WA7）的输出是向最后加法器的输入。

由于第一加法器在树中没有更低的加法器，所以提供基本输入-1，这是为了当在寄存器堆中存储扩展结构中的数据时保证正确寻址。小型整数加法器树基于MAP寄存器中的条目来生成寄存器堆目的地（写）地址。所示特定示例是9个条目的寄存器堆，并且加法器树的基地址被设置成-1。在更大寄存器堆（多于9个条目）的情况下，可以通过将加法器树的基地址设置成基_地址-1来从基地址开始解压缩3×3矩阵。在后一种情况下，3×3矩阵将从指定的基地址而不是位置0开始位于寄存器堆中。

此外，提供来自加法器树中的最后加法器的输出作为向（AGU\DMA）控制器表明需要从存储器获取多少非零值的字节_取回值。将在数据总线（数据_输入）上以按序方式加载这些字节。在通过更宽总线按对或者4个一组发送数据的情况下，将从字节_取回值删除（并且上舍入（round up））1个或者2个最低有效位（lsb）以表示将要传送的64位或者128位数据字的数目。

写地址比较器包括2进制补码整数比较器的树以比较写地址（WA）与字指针（wrd_ptr）值，该值是控制器提供的计数器输出并且从零增量

至从加法器树的末尾输出的字节_取回值。

组合器包括 AND（与）门树以将每个写地址比较器输出与它的对应的位图寄存器值相与以生成写使能（WEN）信号以允许数据从数据总线加载到寄存器堆中适当的位置。具体而言，将 wrd_ptr 值与加法器树生成的 WA 地址(输出)进行比较以确定应当在何时和何处将数据锁存(latch)到寄存器堆中。

在位图寄存器中的条目是零的情况（例如 MCB1、MCB3、MCB5、MCB7）下，显然关联 AND 门的输出（WEN1、WEN3、WEN5、WEN7）将不会被使能并且将没有数据被加载到寄存器堆中对应的位置（r1、r3、r5、r7）。出于这种原因，在时序图中没有示出它们。作为该处理中的初始步骤可以清除寄存器堆（设置成全零值）。当比较器值匹配并且对应的 MCB 位被设置成 1 时，寄存器堆 WEN（写使能）位被设置成高。WEN 位如果被设置则造成数据_输入总线的内容被写入到正确的寄存器-文件寄存器。在所示例子中，当 wrd_ptr 处于零时，第一写使能信号（WEN0）将被使能、即将第一非零元素（A）加载到寄存器堆的位置 r0。类似地，当 wrd_ptr 值处于 1 时，第三写使能信号（WEN2）将被使能，因为 WA2 = MCB2{*1*} + WA1{= *MCB1*(*0*) + *WA0*(*0*)} = 1，因此第二非零数据值（B）将被从数据总线加载到第三寄存器位置 r2。

作为结果，在解压缩\扩展处理结束时，寄存器堆将在其中存储未压缩结构，该未压缩结构在所示例子中将表示为 A、0、B、0、C、0、D、0、E、0。

将理解这里针对 3×3 压缩矩阵概括的相同的一般原理可以用来支持任何任意 N×M 压缩矩阵以及用于解压缩矩阵条目的任何任意大小的寄存器堆目的地。

可以在压缩路径（压缩电路）中采用类似布置。在图 7 中所示压缩路径的情况下，需要浮点比较器树以将未压缩数据结构中的每个数据值（即每个寄存器堆条目）与 0.0 进行比较并且在寄存器堆条目为非零时将位图条目位设置成 1。在用于 9 个条目的寄存器堆的 IEEE 单精度寄存器条目的示例中，需要 9 个这样的浮点比较器。

如图 7 中所示示例性压缩电路包括 32 位比较器组、压缩位图寄存器、读地址计算器、读地址比较器和组合器。与上述扩展电路一样，这些单元在控制器、例如地址生成单元（AGU）或者直接存储器访问（DMA）控

制器的控制之下。现在将参照图 8 的示例性时序图更具体地说明这些单元的构造和操作。

在示例性布置中，未压缩结构（r0-r8）存储于寄存器堆中，将寄存器中的每个值提供给比较器组中的对应的比较器，在该比较器中进行比较以确定该单独的值为非零值还是零值。

虽然将比较器表示为从寄存器堆读取值，但是可以更有利地具有指示寄存器堆中的哪些条目是零\非零值的零位寄存器。该寄存器可以在数据锁存到寄存器堆的写端口时被填充。该零位寄存器可以有利地用于其它目的（例如加速计算）并且在相同申请人的共同未决申请中更具体地描述。这种方式节省时钟周期，因为在压缩周期开始时无需比较。此外，消除了对具有 9 个读端口以对寄存器堆进行并行比较的要求，该要求在布线方面和在功率方面实施起来都很昂贵。另外，这种方式的优点包括减少了所需要的浮点比较器的数目（在具有向寄存器堆的三个写端口的示例性布置中为 3x）。

应当注意在具有多个加载/存储端口的处理器中，必须复制在具有单个加载/存储端口的处理器的情况下可以共享的压缩/解压缩逻辑，一个压缩/解压缩逻辑用于每个独立的加载/存储端口，无论利用单个共享寄存器堆还是多个独立寄存器堆。

然而为了便于说明，将参照图 7 的布置来讨论操作的模式，其中来自比较器组的各个输出作为输入提供给压缩位图寄存器，该寄存器如上所述是大小足以存储用于压缩\未压缩矩阵结构的图的寄存器。然而，将理解一旦加载位图寄存器，在两种不同布置之间操作方式将相同。作为压缩处理中的第一步骤，比较器输出被加载到压缩位图寄存器中。将理解图 7 中的位图寄存器中所示的值完全是示例性的值。

控制器的动作由压缩逻辑和压缩位图寄存器来控制，并且允许时钟周期使用读地址计算器来生成从基地址开始要存储的字节的数目（字节_存储——下文讨论）。

读地址计算器包括 2 进制补码（在所示例子中为 5 位）整数加法器树以针对片上存储器中的每个条目（寄存器-文件）计算读地址（RA）。树中的每个加法器具有两个输入。第一输入是来自位图寄存器的对应值、即位图寄存器中的第一条目是向第一加法器的输入。类似地，位图寄存器中的最后条目是向树中的最后加法器的输入。向每个加法器的第二输入包括

来自树中的前一加法器的输出，因此第一加法器（RA0）的输出是向第二加法器的输入，并且类似地，第八加法器（RA7）的输出是向最后加法器的输入。

由于第一加法器在树中没有更低的加法器，所以提供基本输入 0（即无输入），这在从寄存器堆读取数据时保证正确寻址。整数加法器树基于 MAP 寄存器中的条目来生成寄存器堆目的地（读）地址。

提供来自加法器树中的最后加法器的输出作为向（AGU\DMA）控制器表明需要在片上或者片外存储器中以压缩结构存储多少非零值的字节_存储值。将经由数据总线（数据_输出）以按序方式存储这些字节。

读地址比较器包括 2 进制补码整数比较器的树以比较读地址（RA）与字指针（rd_ptr）值，该值是控制器提供的计数器输出并且从零增量至从加法器树的末尾输出的字节_存储值。

组合器包括 AND 门树。每个 AND 门将读地址比较器输出与它的对应的位图寄存器值组合以生成读使能（REN）信号以允许在适当位置从寄存器堆提取数据。具体而言，将 rd_ptr 值与加法器树生成的 RA 地址（输出）进行比较以确定应当在何时和何处从寄存器堆读取数据。

在位图寄存器中的条目是零的情况（例如 MCB1、MCB3、MCB5、MCB7）下，显然关联 AND 门的输出（REN1、REN3、REN5、REN7）将不会被使能并且将不会从寄存器堆中的对应位置（r1、r3、r5、r7）读取数据。出于这种原因，在时序图中没有示出它们。当比较器值匹配并且对应的 MCB 位被设置成 1 时，寄存器堆 REN（读使能）位被设置成高。REN 位如果被设置则造成将寄存器堆中所选择条目的内容被放置于数据_输出总线上。在所示例子中，当 rd_ptr 处于零时，第一读使能信号（REN0）将被使能、即将第一非零元素（A）从寄存器堆的位置 r0 读取到数据_输出上。类似地，当 rd_ptr 值处于 2 时，第三读使能信号（REN2）将被使能，因为 $RA2 = MCB2\{1\} + RA\{1\} = 2$，因此第二非零数据值（C）将从第三寄存器位置 r2 被锁存到数据_输出上。

作为结果，在压缩处理结束时，寄存器堆将把控制器在片上或者片外存储器中存储的值 A、C、E、G 和 I 与来自图寄存器的图一起放置于数据_输出上。

将理解这里针对 3×3 未压缩矩阵概括的相同的一般原理可以用来支持任何任意 N×M 未压缩矩阵。

根据以上说明将理解，相同的硬件可以在控制器的总体控制之下用于压缩和解压缩。

在图 9 中示出了这样的组合的布置，该布置还合并有使比较器附着到上文讨论的寄存器堆写端口的布置。在本示例中，连接到寄存器堆的处理器数据路径需要 3 个写端口。在实践中，提出的方案可以容易地扩展到任意数目的寄存器堆写端口和具有任意数目的条目的寄存器堆。将理解 wr_ptr 和 rd_ptr 和其它信号将按照需要由控制器操作/接收以压缩数据以供片上或者片外存储或者在将数据从片上或者片外储存器加载到寄存器堆中时扩展数据。

为了便于说明，没有示出基_地址-1。

可能需要附加执行流水线级以便保证回写（write-back）浮点比较器的延迟没有与处理器的执行路径串行发生，因为这否则可能限制最大时钟速率并且因此限制处理器的 FLOPS 速率。

在本申请中描述的新方法提供多个显著优点，这些优点例如包括通过减少存储平凡值所需要的存储器来减少处理器所需的存储器带宽、减少为了在带宽方面高效地通过片上或者片外总线向存储器和从存储器移动无论已压缩或者未压缩的标量、矢量和矩阵数据而需要的总线大小、减少功率耗散、增加处理器在从存储器或者处理器寄存器读取时对已压缩标量、矢量和矩阵数据进行操作时的有效处理能力（FLOPS），并且减少了处理器的延时。

在附图中将理解所示的线可以与多条线相对应并且这是由划过线和相邻的数字来代表的，因此例如图 9 中的存储_数据总线代表 32 位总线，而加法器树的每个输出是 5 位。

在本说明书中使用时的措辞包括（comprises/comprising）是为了指明存在声明的特征、整数、步骤或者组成，但是并不排除存在或者增加一个或者多个其它特征、整数、步骤、组成或者其组合。

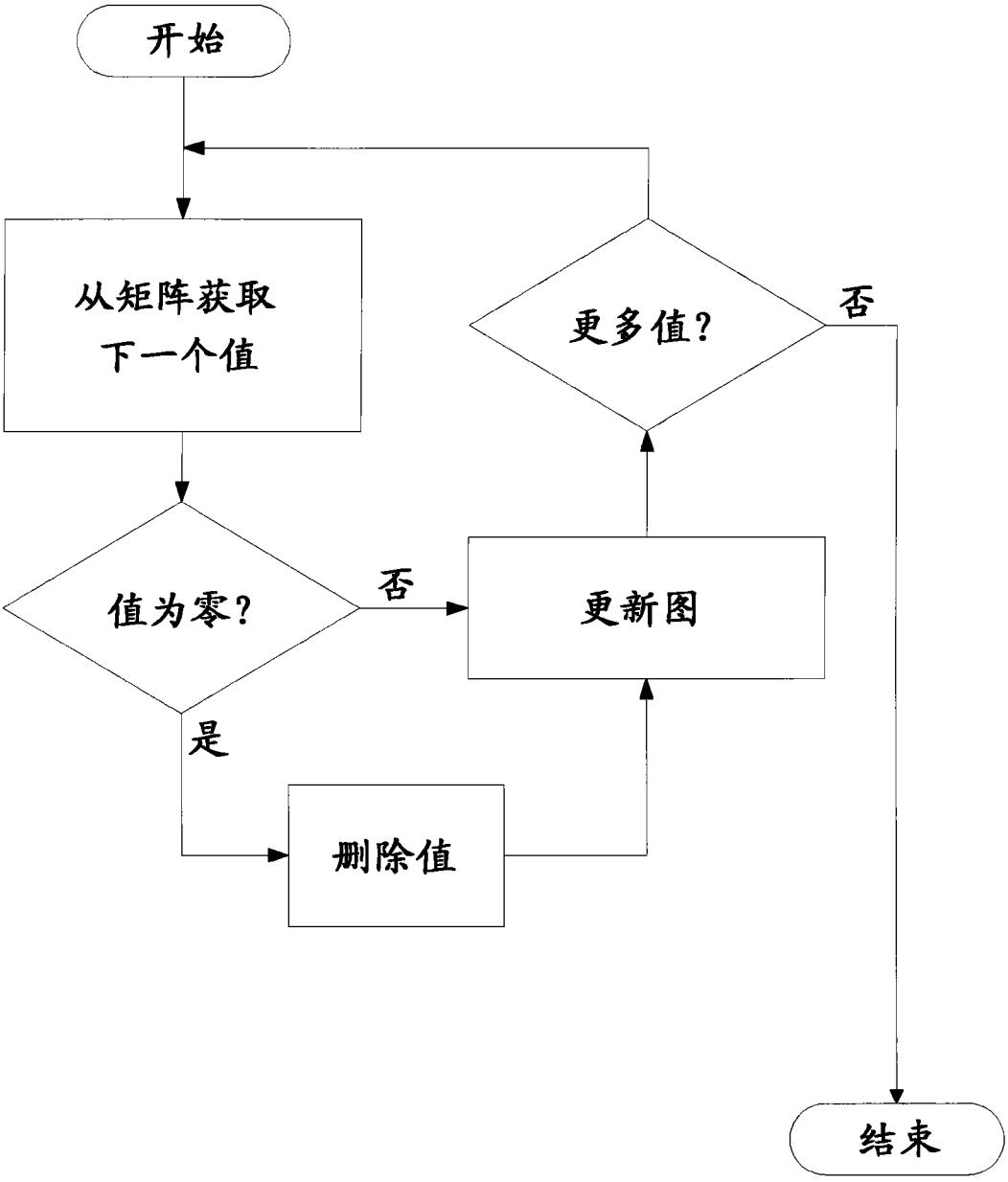


图1

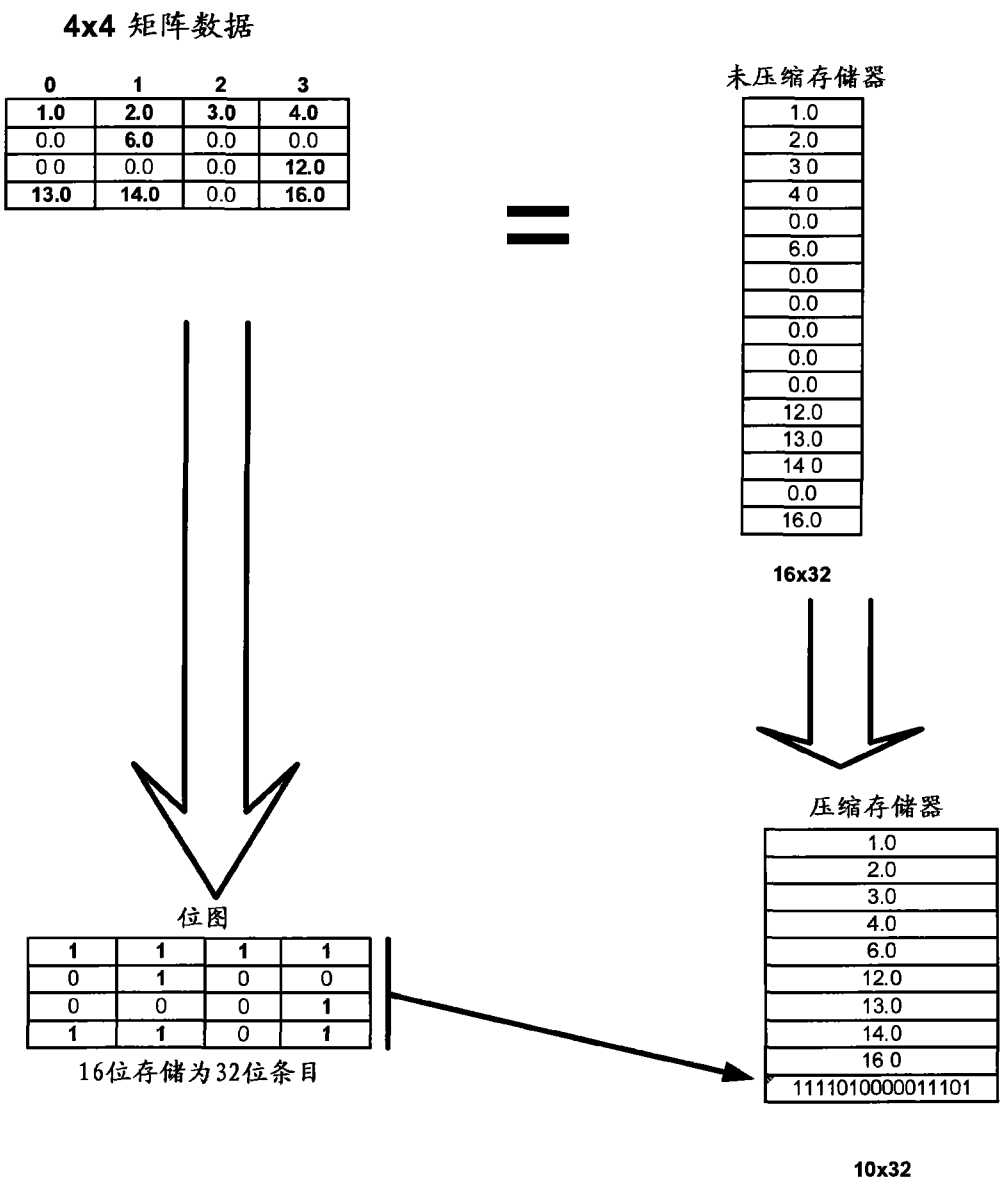

4x4 矩阵数据
0 1 2 3
1.0 2.0 3.0 4.0
0.0 6.0 0.0 0.0
0 0 0.0 0.0 12.0
13.0 14.0 0.0 16.0
=
未压缩存储器
1.0
2.0
3 0
4 0
0.0
6.0
0.0
0.0
0.0
0.0
0.0
12.0
13.0
14 0
0.0
16.0
16x32
压缩存储器
1.0
2.0
3.0
4.0
6.0
12.0
13.0
14.0
16 0
1111010000011101
10x32
位图
1 1 1 1
0 1 0 0
0 0 0 1
1 1 0 1
16位存储为32位条目

图2

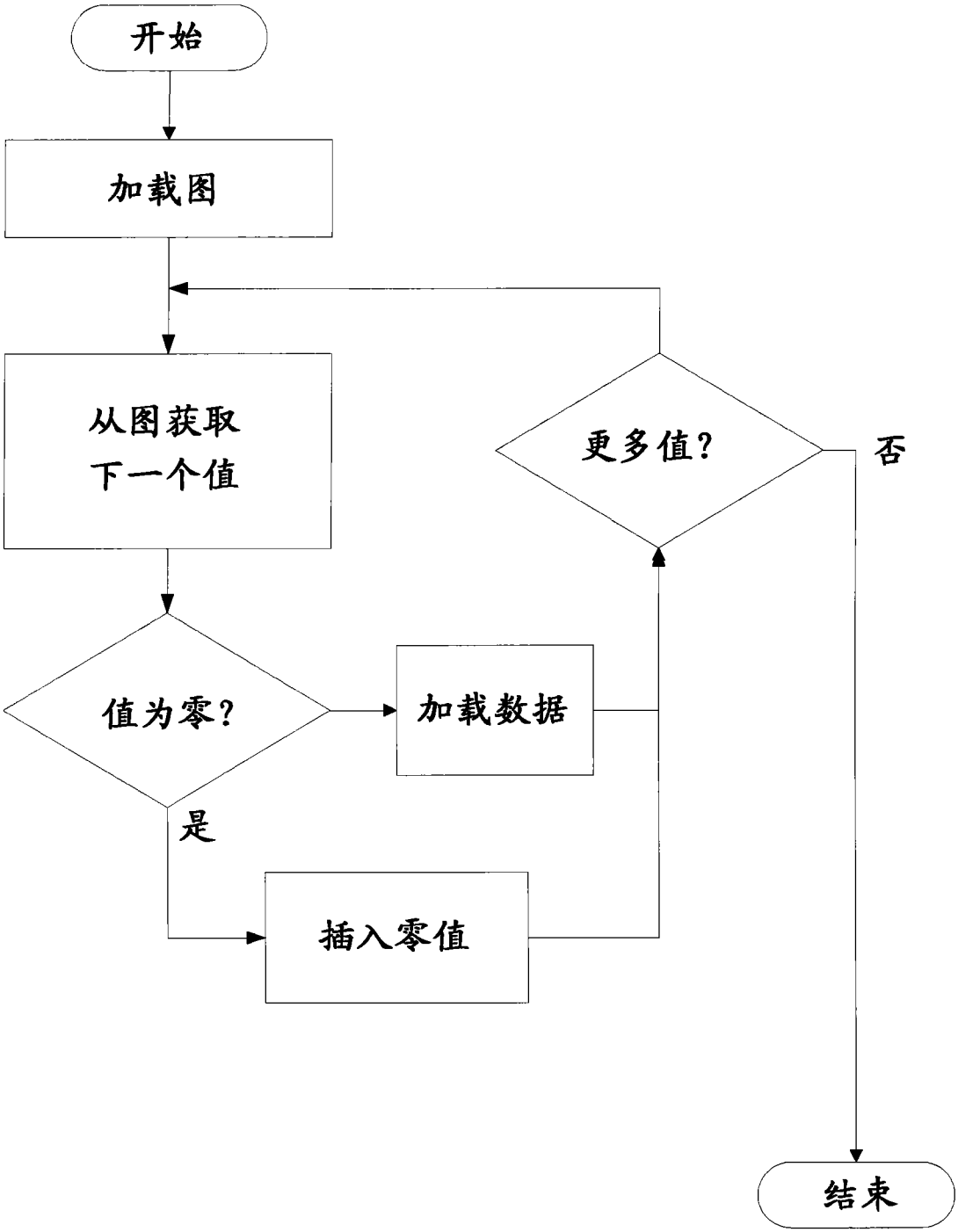


图3

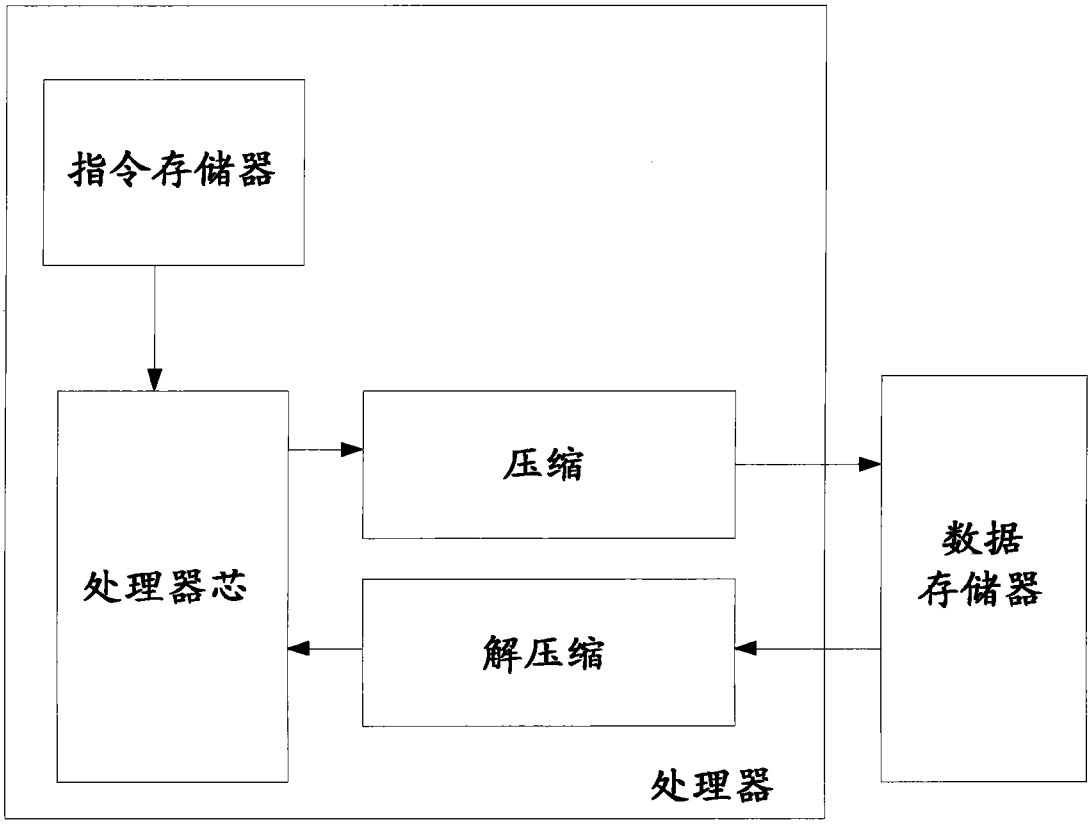


图4

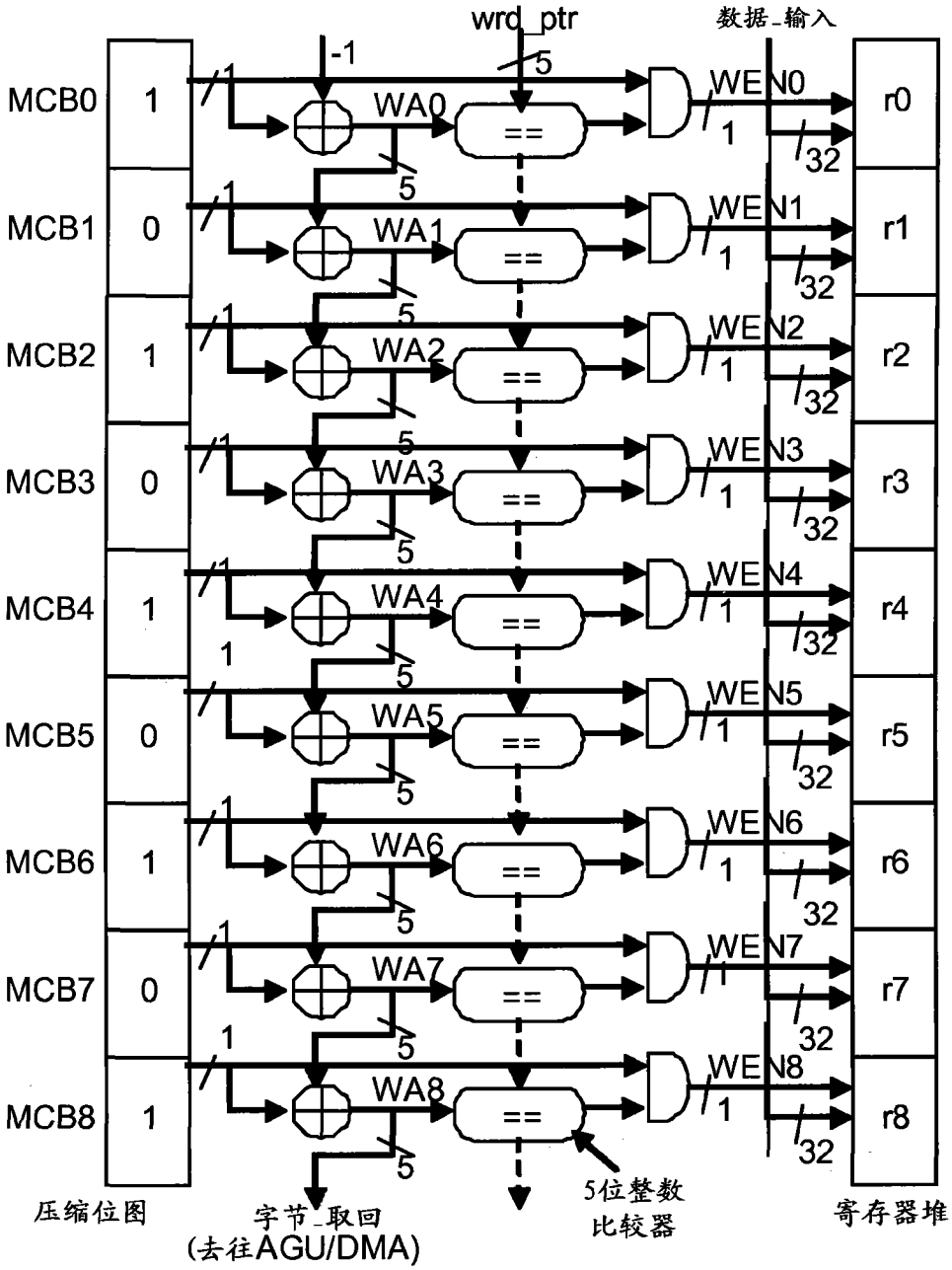

wrq_ptr
数据_输入
-1
5
MCB0
MCB1
MCB2
MCB3
MCB4
MCB5
MCB6
MCB7
MCB8
1
0
1
0
1
0
1
0
1
WA0
WA1
WA2
WA3
WA4
WA5
WA6
WA7
WA8
==
WEN0
WEN1
WEN2
WEN3
WEN4
WEN5
WEN6
WEN7
WEN8
r0
r1
r2
r3
r4
r5
r6
r7
r8
压缩位图
字节_取回
(去往AGU/DMA)
5位整数
比较器
寄存器堆

图5

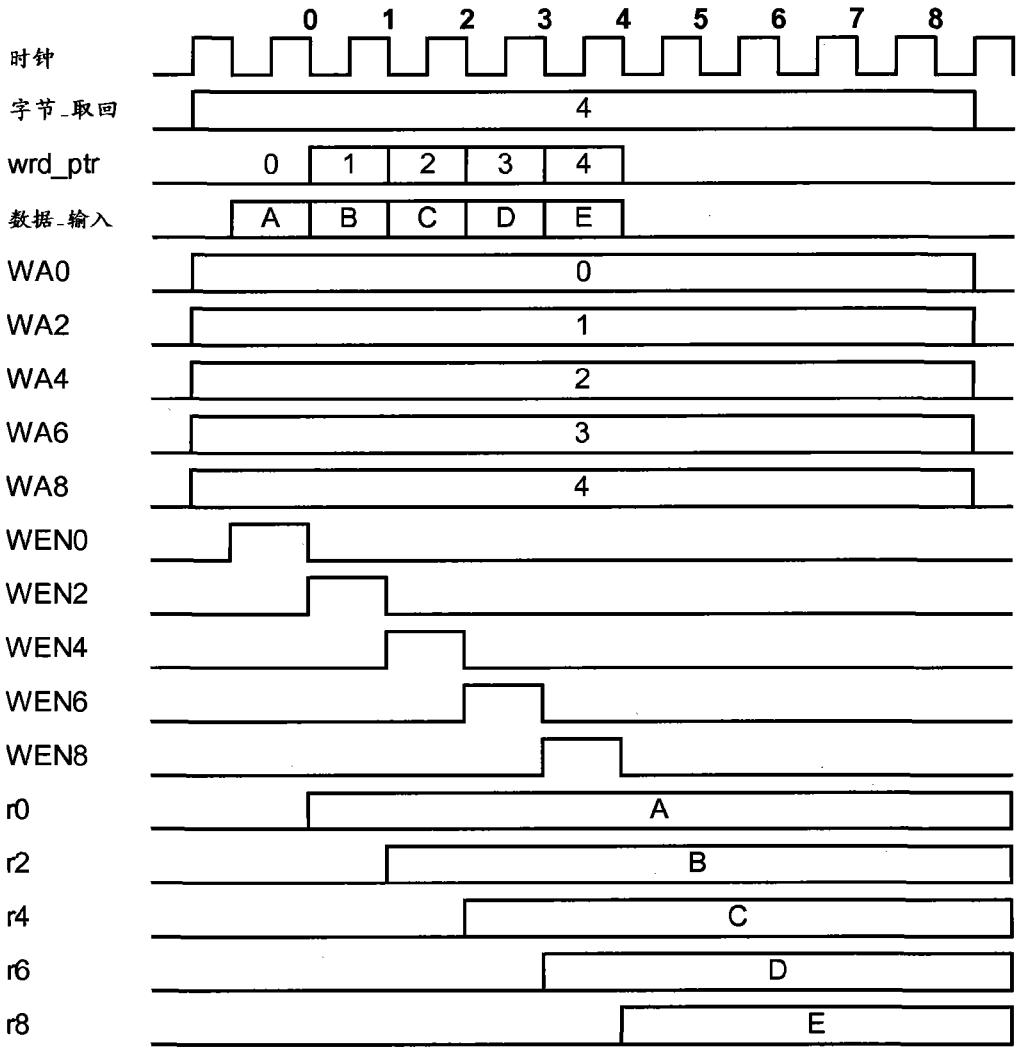

0 1 2 3 4 5 6 7 8
时钟
字节_取回
4
wrd_ptr
0 1 2 3 4
数据_输入
A B C D E
WA0
0
WA2
1
WA4
2
WA6
3
WA8
4
WEN0
WEN2
WEN4
WEN6
WEN8
r0
A
r2
B
r4
C
r6
D
r8
E

图6

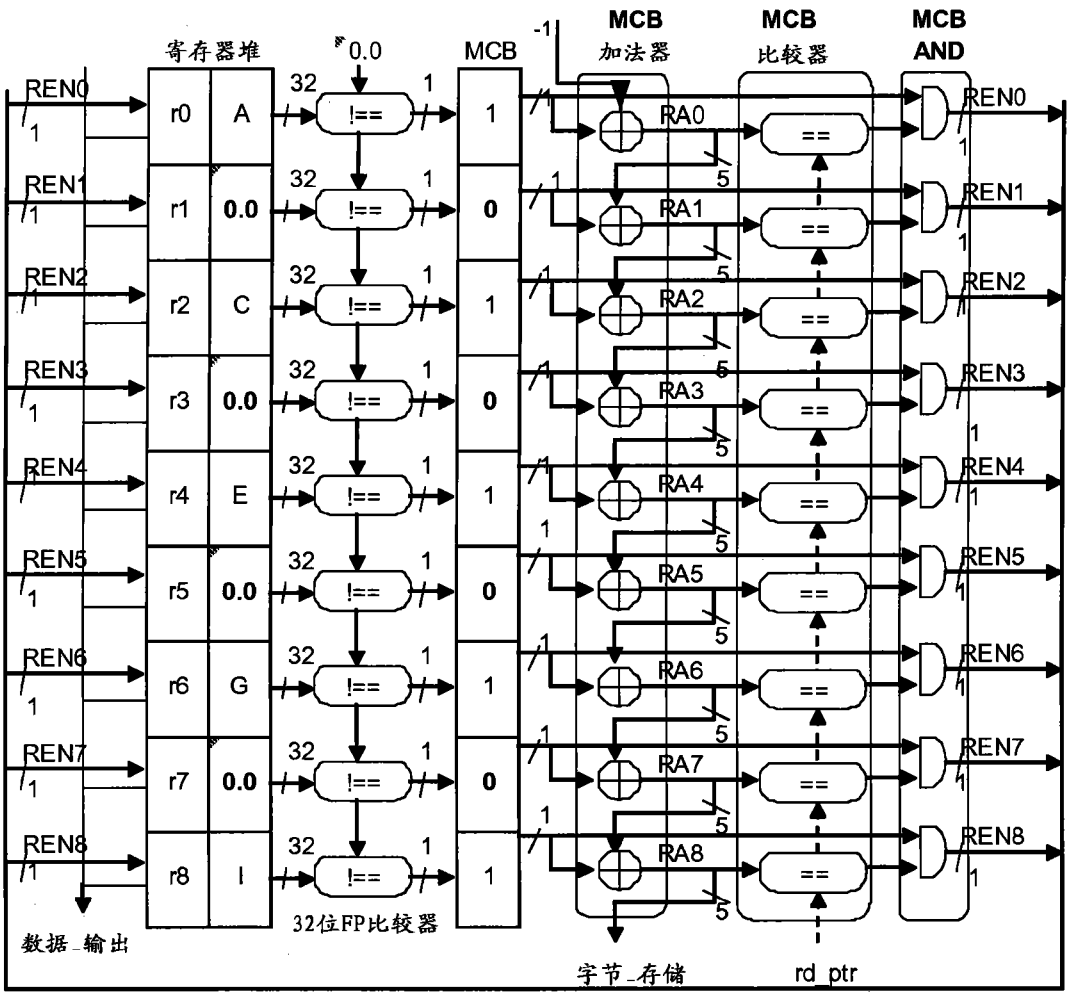

寄存器堆
0.0
MCB
MCB
加法器
MCB
比较器
MCB
AND
-1
REN0
REN1
REN2
REN3
REN4
REN5
REN6
REN7
REN8
r0
r1
r2
r3
r4
r5
r6
r7
r8
A
0.0
C
0.0
E
0.0
G
0.0
I
32
!==
1
RA0
RA1
RA2
RA3
RA4
RA5
RA6
RA7
RA8
5
==
数据_输出
32位FP比较器
字节_存储
rd_ptr

图7

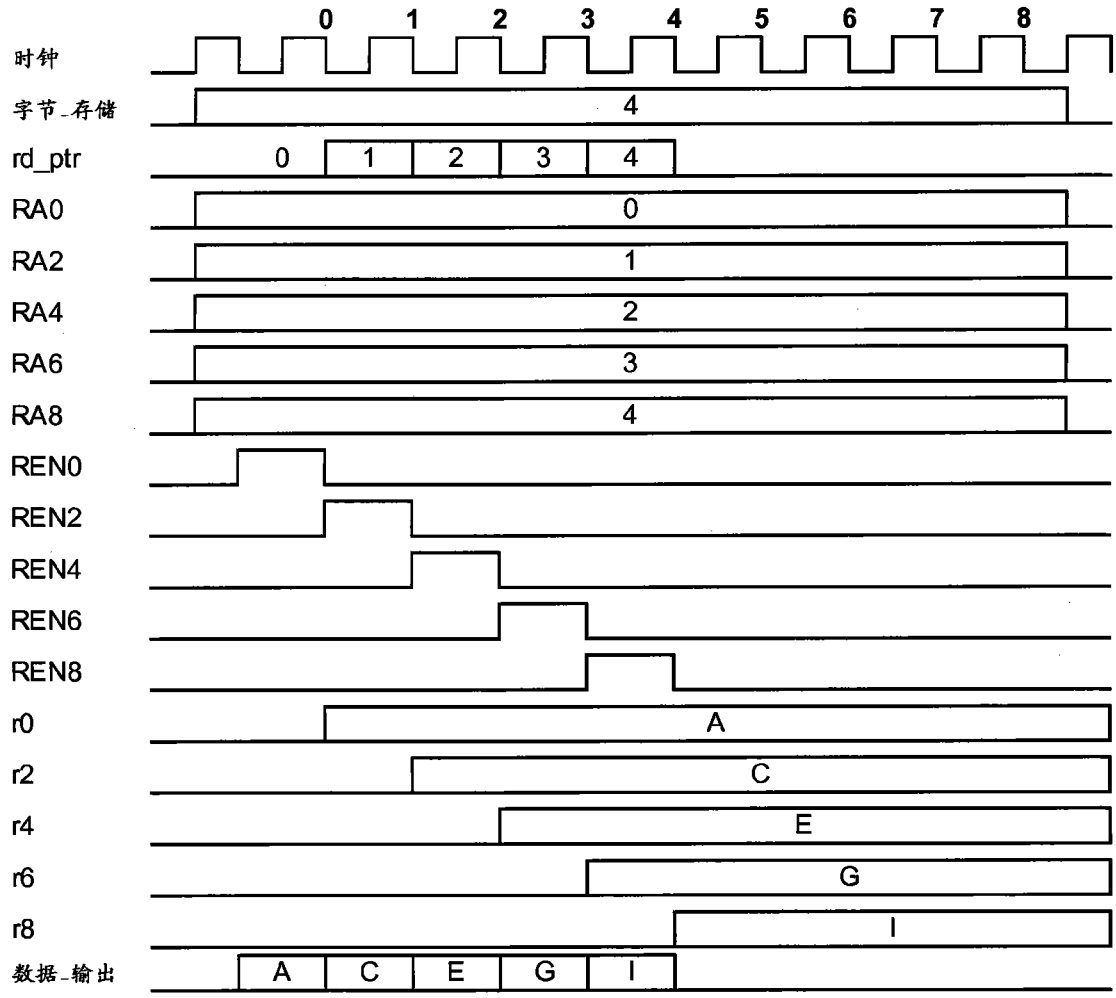

0 1 2 3 4 5 6 7 8
时钟
字节_存储
4
rd_ptr
0 1 2 3 4
RA0
0
RA2
1
RA4
2
RA6
3
RA8
4
REN0
REN2
REN4
REN6
REN8
r0
A
r2
C
r4
E
r6
G
r8
I
数据_输出
A C E G I

图8

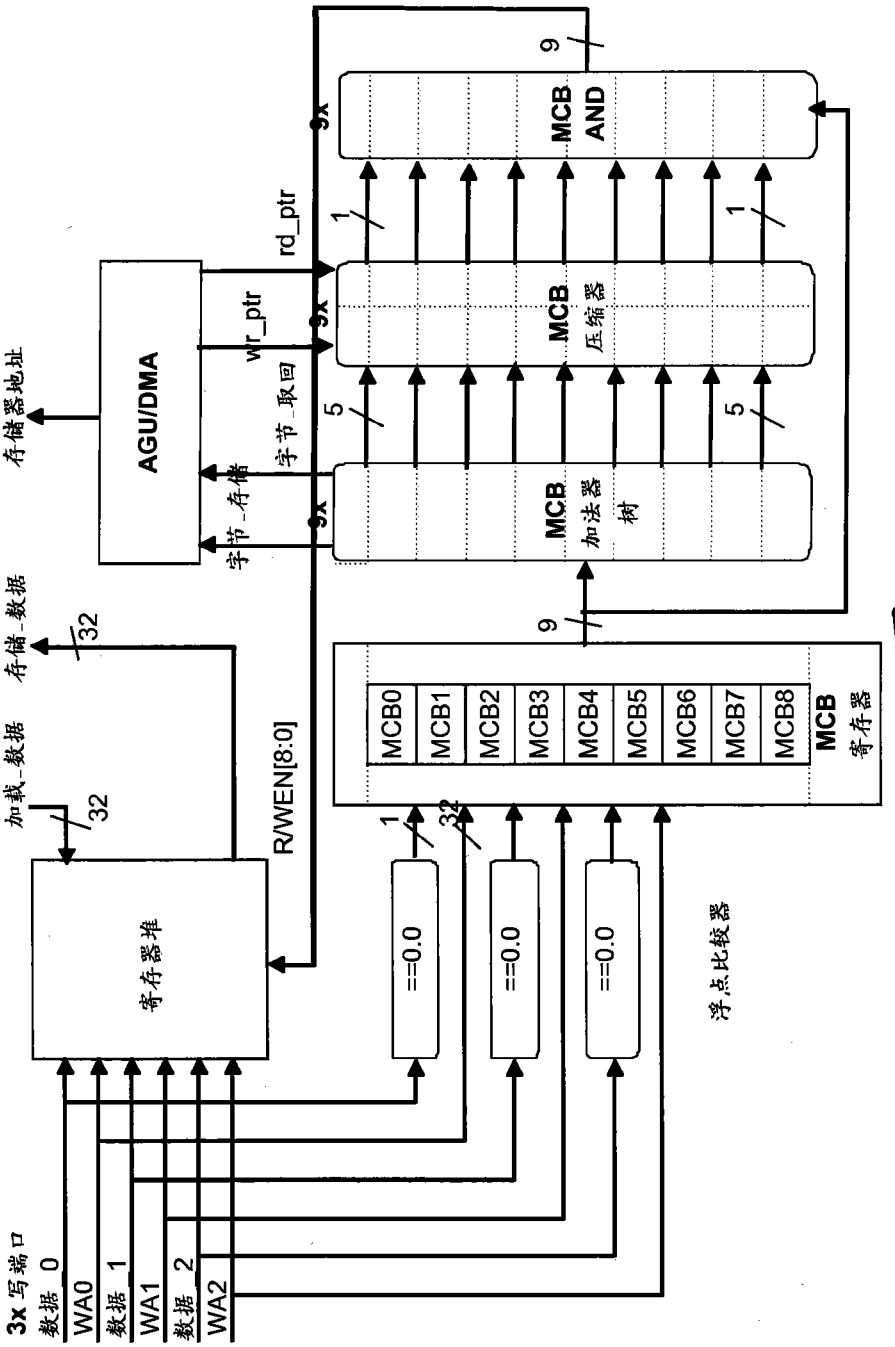

3x 写端口
数据_0
WA0
数据_1
WA1
数据_2
WA2
寄存器堆
加载_数据
32
存储_数据
32
存储器地址
AGU/DMA
R/WEN[8:0]
wr_ptr
rd_ptr
字节_存储
字节_取回
9x
9x
9x
MCB 加法器树
MCB 压缩器
MCB AND
MCB 寄存器
MCB0
MCB1
MCB2
MCB3
MCB4
MCB5
MCB6
MCB7
MCB8
==0.0
==0.0
==0.0
浮点比较器
1
5
9
32

图 9