[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H03M 13/25
H04L 27/34 H04L 27/26

[12] 发明专利申请公开说明书

[21] 申请号 02820735.1

[43] 公开日 2005 年 1 月 26 日

[11] 公开号 CN 1572060A

[22] 申请日 2002.8.23 [21] 申请号 02820735.1

[30] 优先权

[32] 2001. 8.23 [33] US [31] 60/314,168

[32] 2001. 8.23 [33] US [31] 60/314,169

[86] 国际申请 PCT/CA2002/001297 2002.8.23

[87] 国际公布 WO2003/019791 英 2003.3.6

[85] 进入国家阶段日期 2004.4.19

[71] 申请人 北方电讯网络有限公司

地址 加拿大魁北克省

[72] 发明人 童 文 A·坎达尼

[74] 专利代理机构 中国专利代理(香港)有限公司

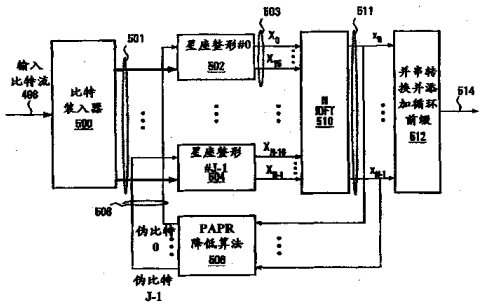
代理人 杨 凯 王 勇

权利要求书 8 页 说明书 22 页 附图 12 页

[54] 发明名称 降低多载波调制中的峰值功率的系统和方法

[57] 摘要

提出了一种用于降低峰值对平均功率比的多谐波调制方法。该方法包括针对每个中心频率定义相应的信号星座(它具有针对每个输入元素集合的冗余映射)；然后选择可使多谐波信号的峰值功率降低的特定映射排列。还采用了一种定义了块的分层结构的整形方法，其中各层具有的块比各前一层少。该分层结构用于选择扩展星座的特定子集。



ISSN1008-4274

1. 一种多谐波调制方法，它包括如下步骤：

针对多个中心频率中的每一个中心频率定义包括多个星座点的相应的信号星座；

针对每个谐波和 M 个数据元素值的每种排列，定义具有至少一个星座点的相应集合，该集合中的任何一个星座点可用于表示 M 个数据元素值的排列，从而提供用于表示 M 个数据元素值的排列的可供选择的星座点；至少一个具有至少一个星座点的所述集合具有一个以上的星座点；

通过将数据元素映射到所述多个信号星座中的各信号星座来生成多谐波信号，在有可供选择的星座点的情况下，选择可使所述多谐波信号的峰值对平均功率比降低的星座点。

2. 如权利要求 1 所述的方法，其特征在于，在有可供选择的星座点的情况下选择可使所述多谐波信号的峰值功率降低的星座点的步骤包括：搜索可选星座点的所有可能排列，以获得可使可能峰值对平均功率比绝对最小的星座点集合。

3. 如权利要求 1 所述的方法，其特征在于，在有可供选择的星座点的情况下，选择可使所述多谐波信号的峰值功率降低的星座点的步骤包括：搜索可选星座点的可能排列，以获得可使峰值对平均功率比小于预定阈值的星座点集合。

4. 如权利要求 1 所述的方法，其特征在于，具有至少一个星座点的每个集合包含两个星座点。

5. 一种用于将 M×K 个数据元素集调制到多谐波信号上的方法，其中将 K 个数据元素调制到 M 个中心频率中的每一个中心频率上；K >=1 和 M >=2，所述方法包括：

针对所述 M 个中心频率中的每一个中心频率和 K 个数据元素值的每种排列，定义相应的多个星座点，其中所述多个星座点可用于表示该中心频率上 K 个数据元素值的排列；

对于 M×K 个数据元素的特定集合，针对所述 M 个中心频率中的每一个中心频率定义 K 个数据元素值的特定排列；针对所述 M 个中心频率中的每一个中心频率

确定特定的不同的多个星座点;

从每组特定的不同的多个星座点中选择一个可使所述多谐波信号的峰值对平均功率比降低的星座点。

6. 如权利要求5所述的方法，其特征在于，所述方法还包括如下步骤:

在选择所述单星座点时，搜索每组特定的不同的多个星座点中单个星座点的所有可能组合，以使所述多谐波信号的峰值对平均功率比为最小可能值。

7. 如权利要求5所述的方法，其特征在于，所述方法还包括如下步骤:

在选择所述单星座点时，搜索每组特定的不同的多个星座点中单星座点的组合，以使所述多谐波信号的峰值均值功率小于预定阈值。

8. 如权利要求7所述的方法，其特征在于，搜索单星座点的组合包括如下步骤:

随机地从每组特定的不同的多个星座点中选择一个星座点;

利用所述随机选择的星座点估算峰值对平均功率比;

如果所述峰值对平均功率比小于预定阈值，则停止所述搜索;

重复执行如下步骤直到所述峰值对平均功率比小于预定阈值为止:将所述随机选择的星座点中的至少一个改为所述随机选择的星座点中的至少一个所属的所述特定的不同的多个星座点中的另一个星座点，并重新估算峰值对平均功率比。

9. 如权利要求1所述的方法，其特征在于，所述方法还包括:针对各信号星座，通过如下步骤确定用于表示所述M个数据元素值的给定排列的所述至少一个星座点:

针对所述M个数据元素和至少一个伪数据元素的每种排列执行星座整形;对于各信号星座，针对伪数据元素值的每种排列执行星座整形以生成相应的经整形的输出，给定信号星座的相应的整形输出共同包括用于表示所述M个数据元素值排列的至少一个星座点。

10. 如权利要求9所述的方法，其特征在于，对包括多个星座点的给定信号星座执行星座整形的步骤包括:

将某个成本与所述多个星座点中的每个星座点相关联;

定义块的分层结构;所述分层结构具有包括至少第一和最后一层的多个层;各

层具有的块比其前一层少;

其中所述第一层是通过如下步骤形成的:按照成本将所有星座点排序,然后将最低成本的星座点组分配给第一整形分割,如果划分包括多个整形分割,则继续将次低成本的星座点组分配给第二整形分割,并依此类推直到将最高成本的星座点组分配给最后一个整形分割,其中根据各整形分割中星座点的成本来为该整形分割分配成本,以及每个整形分割是第一层块;

其中,其他每个层上的元素是通过合并前一层的两个块形成的,且其成本根据该前一层的两个块的成本来进行分配,每一层的块根据成本由一个所述元素或一组所述元素组成;

所述最后一层只有一个包括多个元素的块;

整形增益通过只映射到所述最后一层的元素的子集来取得。

11. 一种针对包括多个星座点的信号星座执行星座整形的方法,它包括如下步骤:

将某个成本与所述多个星座点中的各星座点相关联;

定义块的分层结构;所述分层结构具有包括至少第一和最后一层的多个层;各层具有的块比其前一层少;

其中所述第一层是通过如下步骤形成的:按照成本将所有星座点排序,然后将最低成本的星座点组分配给第一整形分割,如果划分包括多个整形分割,则继续将次低成本的星座点组分配给第二整形分割,并依此类推直到将最高成本的星座点组分配给最后一个整形分割,其中根据各整形分割中星座点的成本来为该整形分割分配成本,以及每个整形分割是第一层块;

其中,其他每个层上的元素是通过合并前一层的两个块形成的,且其成本根据该前一层的两个块的成本来进行分配,每一层的块根据成本由一个所述元素或一组所述元素组成;

所述最后一层只有一个包括多个元素的块;

整形增益通过只映射到所述最后一层的元素的子集来取得。

12. 如权利要求 11 所述的方法,其特征在于,分配给各星座点的成本是其能量。

13. 如权利要求 12 所述的方法，其特征在于，至少一些所述层块只是前一层块的重新排序合并。

14. 如权利要求 13 所述的方法，其特征在于，在其中各块是前一层块重新排序合并成的块组的层上，所有组都具有相同的大小。

15. 如权利要求 13 所述的方法，其特征在于，在其中各块是前一层块重新排序合并成的块组的至少一层上，所述组具有不同的大小。

16. 如权利要求 11 所述的方法，其特征在于，所述方法还包括如下步骤：

通过如下步骤执行寻址：

应用输入比特集合的第一子集来识别所述最高层的块中的元素；

应用所述输入比特集合的后续子集来识别后续层上的块，其中在所述第一层识别特定的整形分割；

在所述第一层，应用输入比特的最后一个子集来识别所述第一层上识别的所述特定整形分割中的特定的信号星座点。

17. 如权利要求 16 所述的方法，其特征在于，采用基于霍夫曼树的寻址方法。

18. 如权利要求 16 所述的方法，其特征在于，采用基于固定树的寻址方法。

19. 如权利要求 11 所述的方法，其特征在于：采用了 256 个点的星座，其中层 1 有 16 个分割，各包括 16 个星座点；层 2 有 8 个块，各包括 16 个元素；层 3 有 4 个块，各包括 8 个元素，这 8 个元素的大小为{16、16、32、32、32、32、32、64}；层 4 有 2 个块，各包括 64 个元素；以及层 5 只有一个块，包括 4096 个元素。

20. 如权利要求 16 所述的方法，其特征在于，所述输入比特包括多个数据比特和至少一个伪比特，所述方法还包括如下步骤：

针对所述至少一个伪比特的值的每种排列重复执行如权利要求 16 所述的方法。

21. 如权利要求 20 所述的方法，其特征在于，所述方法还包括如下步骤：

针对多个信号星座中的每一个重复执行权利要求 20 所述的方法，以生成每个信号星座的相应的多个整形输出；

通过从各相应的多个整形输出中适当地选择一个可使峰值平均功率降低的整形输出。

22. 一种多谐波调制器，它包括:

信号发生器，适于针对多个中心频率中的每一个中心频率生成对应包括多个星座点的相应信号星座的信号;

星座映射器，适于针对每个谐波和 M 个数据元素值的每种排列定义具有至少一个星座点的相应集合，其中任何一个星座点可用于表示 M 个数据元素值的排列，从而提供可选择用于表示 M 个数据元素值排列的星座点，且至少一个具有至少一个星座点的所述集合具有一个以上的星座点;

峰值平均功率降低（PAPR 功能），适于针对每个中心频率从由所述星座映射器映射到的至少一个星座点中选择可使所述多谐波信号的峰值对平均功率比降低的星座点，其中，利用所述峰值平均功率降低功能选择的星座点生成信号并发送。

23. 如权利要求 22 所述的调制器，其特征在于，所述 PAPR 功能适于在有可供选择的星座点的情况下，通过搜索可选星座点的所有可能排列以获得使可能峰值对平均功率比绝对最小的星座点集合，从而选择可使所述多谐波信号的峰值功率降低的星座点。

24. 如权利要求 22 所述的调制器，其特征在于，所述 PAPR 功能适于在有可供选择的星座点的情况下，通过搜索可选星座点的所有可能排列以获得可使峰值对平均功率比小于预定阈值的星座点集合，从而选择可使所述多谐波信号的峰值功率降低的星座点。

25. 如权利要求 22 所述的调制器，其特征在于它还包括针对各信号星座的相应的星座整形器，该星座整形器适于通过如下步骤确定用于表示所述 M 个数据元素值的给定排列的至少一个星座点:

针对 M 个数据元素和至少一个伪数据元素的每种排列执行星座整形；对各信号星座，针对伪数据元素值的每种排列执行所述星座整形，以生成相应的整形输出;给定信号星座的相应的整形输出共同包括用于表示所述 M 个数据元素值排列的至少一个星座点。

26. 如权利要求 25 所述的调制器，其特征在于，所述星座整形器适于通过如下步骤针对包括多个星座点的给定信号星座执行星座整形:

将所述多个星座点中的各星座点与某个成本相关联;

定义块的分层结构，所述分层结构具有包括至少第一和最后一层的多个层，各层具有的块比其前一层少；

其中所述第一层是通过如下步骤形成的：按照成本将所有星座点排序，然后将最低成本的星座点组分配给第一整形分割，如果划分包括多个整形分割，则继续将次低成本的星座点组分配给第二整形分割，并依此类推直到将最高成本的星座点组分配给最后一个整形分割，其中根据各整形分割中星座点的成本来为该整形分割分配成本，以及每个整形分割是第一层块；

其中,其他每个层上的元素是通过合并前一层的两个块形成的，且其成本根据该前一层的两个块的成本来进行分配，每一层的块根据成本由一个所述元素或一组所述元素组成；

所述最后一层只有一个包括多个元素的块；

整形增益通过只映射到所述最后一层的元素的子集来取得。

27. 如权利要求 26 所述的调制器，其特征在于，至少一些所述层块只是前一层块的重新排序合并。

28. 如权利要求 26 所述的调制器，其特征在于，在其中各块是前一层块重新排序合并成的块组的层上，所有组具有相同的大小。

29. 如权利要求 26 所述的调制器，其特征在于，在其中各块是前一层块重新排序合并成的块组的至少一层上，所述组具有不同的大小。

30. 如权利要求 26 所述的调制器，其特征在于，所述调制器还适于通过如下步骤执行寻址：

应用输入比特集合的第一子集来识别所述最高层的块中的元素；

应用所述输入比特集合的后续子集来识别后续层上的块，其中在所述第一层识别特定的整形分割；

在所述第一层，应用输入比特的最后一个子集来识别所述第一层上识别的所述特定整形分割中的特定的信号星座点。

31. 如权利要求 30 所述的调制器，其特征在于，采用基于霍夫曼树的寻址方法。

32. 如权利要求 30 所述的调制器，其特征在于，采用基于固定树的寻址方法。

33. 如权利要求 30 所述的调制器，其特征在于：采用了 256 个点的星座，其中层 1 有 16 个分割，各包括 16 个星座点；层 2 有 8 个块，各包括 16 个元素；层 3 有 4 个块，各包括 8 个元素，这 8 个元素的大小为{16、16、32、32、32、32、32、64}；层 4 有 2 个块，各包括 64 个元素；以及层 5 只有一个块，包括 4096 个元素。

34. 一种调制器，它包括：

整形装置，用于针对多个信号星座中的每一个多次执行星座整形，以生成相应的多个整形输出；

峰值平均功率降低装置，用于从每组相应的整形输出中选择相应的一个整形输出，以使通过合并所述各单个整形输出而形成的信号的峰值功率降低。

35. 一种其中存储了用以命令处理平台降低峰值功率的指令的计算机可读媒体，所述指令包括：

这样的代码段，用以针对多个中心频率中的每一个中心频率定义包括多个星座点的相应的信号星座；

这样的代码段，用以针对每个谐波和 M 个数据元素值的每种排列，定义具有至少一个星座点的相应集合，其中任何一个星座点可用于表示 M 个数据元素值的排列，从而提供用于表示 M 个数据元素值的排列的可供选择的星座点，其中至少一个具有至少一个星座点的所述集合具有一个以上的星座点；

这样的代码段，用于以如下步骤来生成多谐波信号：将数据元素映射到所述多个信号星座中的每个信号星座，并且在有可供选择的星座点的情况下，选择可使所述多谐波信号的峰值对平均功率比降低的星座点。

36. 一种其中存储了用以指令处理平台针对包括多个星座点的信号星座执行星座整形的指令的计算机可读媒体，所述指令包括：

这样的代码段，用以将某个成本与所述多个星座点中的各星座点相关联；

这样的代码段，用以定义块的分层结构，所述分层结构具有包括至少第一和最后一层的多个层，各层具有的块比其前一层少；

其中所述第一层是通过如下步骤形成的：按照成本将所有星座点排序，然后将最低成本的星座点组分配给第一整形分割，如果划分包括多个整形分割，则继续将次低成本的星座点组分配给第二整形分割，并依此类推直到将最高成本的星座点组

分配给最后一个整形分割，其中根据各整形分割中星座点的成本来为该整形分割分配成本，以及每个整形分割是第一层块;

其中,其他每个层上的元素是通过合并前一层的两个块形成的,且其成本根据该前一层的两个块的成本来进行分配，每一层的块根据成本由一个所述元素或一组所述元素组成;

所述最后一层只有一个包括多个元素的块;

整形增益通过只映射到所述最后一层的元素的子集来取得。

37. 一种适于实现如权利要求 11 所述方法的处理平台。

降低多载波调制中的峰值功率的系统和方法

发明领域

本发明涉及多载波调制系统和方法。

发明背景

用于多谐波调制的常规系统和方法针对多个中心频率中的每一个中心频率采用信号星座，其中唯一地表示数据元素（如比特）的每种可能排列。

多谐波信号包括调制到每个中心频率信号星座上的数据元素。这种调制可以按频率来执行，也可以采用例如 IFFT 技术以并行方式执行。

不利的方面在于，频率星座点的某些排列会导致多谐波信号中的高峰值功率。频率星座点的其他排列会导致多谐波信号中的低峰值功率。

图 1 中显示了此情况的一个实例，其中标号 10 处所示的是例如 OFDM 中可能采用的多载波配置。在此实例中，有八个载波。多载波信号的振幅通过将这些多载波所用星座点以复数形式相加得到。通常，这种复数加法将会得到靠近某个平均值的值，因为正在合并的值通常具有某种程度上的随机性。某种程度上随机的复数加法的实例如标号 12 处所示，而合并的振幅如标号 14 处所示。

在其他情况中，复数加法将会导致很大的峰值，因为这些值全部以相加方式合并。此情况的实例如标号 16 处所示，其中合并的振幅如 18 所示。

最好有一种不存在上述这些缺点的多载波调制方法。

发明概述

本发明的一个宽的方面提供了一种多谐波调制方法。该方法包括针对多个中心频率中的每一个中心频率定义包括多个星座点的相应的信号星座；针对每个谐波和 M 个数据元素值的每种排列，定义具有至少一个星座点的相应集合（其中任何一个星座点可用于表示 M 个数据元素值的排列），从而提供用于表示 M

个数据元素值的排列的可供选择的星座点；至少一个具有至少一个星座点的所述集合具有一个以上的星座点；通过如下步骤来生成多谐波信号：将数据元素映射到多个信号星座，并且在有可供选择的星座点的情况下，选择可使多谐波信号的峰值对平均功率比降低的星座点。

在一些实施例中，在有可供选择的星座点的情况下，选择使所述多谐波信号的峰值功率降低的星座点的步骤包括：搜索可选星座点的所有可能排列，以获得使可能的峰值对平均功率比绝对最小的星座点集合。

在一些实施例中，在有可供选择的星座点的情况下，选择使所述多谐波信号的峰值功率降低的星座点的步骤包括：搜索可选星座点的所有可能排列，以获得使可能峰值对平均功率比小于预定阈值的星座点集合。

在一些实施例中，具有至少一个星座点的每个集合包括两个星座点。

本发明的另一个宽的方面是提供一种用于将 M×K 个数据元素集调制到多谐波信号上的方法；其中将 K 个数据元素调制到 M 个中心频率中的每一个中心频率上，K >=1 和 M >=2。该方法包括，针对 M 个中心频率中的每一个中心频率和 K 个数据元素值的每种排列，定义相应的可用于表示该中心频率上 K 个数据元素值的排列的多个星座点；对于 M×K 个数据元素的特定集合，针对 M 个中心频率中的每一个中心频率定义 K 个数据元素值的特定排列；针对 M 个中心频率中的每一个中心频率确定特定的不同的多个星座点；从每组特定的不同的多个星座点中选择一个星座点，以使多谐波信号的峰值对平均功率比降低。

在一些实施例中，搜索单个星座点的组合包括随机地从每个特定的不同的多个星座点中选择星座点；利用随机选择的星座点估算峰值对平均功率比；如果峰值对平均功率比小于预定阈值，则停止搜索；重复执行如下步骤直到峰值对平均功率比小于预定阈值为止：将随机选择的星座点中的至少一个改为该其所属特定的相应星座点集合中的另一个星座点，并重新估算峰值对平均功率比。

在一些实施例中，针对各信号星座，确定用于表示 M 个数据元素值给定排列的至少一个星座点包括针对 M 个数据元素和至少一个伪数据元素的每种排列执行星座整形；对于各信号星座，针对伪数据元素值的每种排列执行星座整形，以生成相应的整形输出；给定信号星座的相应的整形输出共同包括用于表示 M

个数据元素值排列的至少一个星座点。

本发明的另一个宽的方面提供一种对包括多个星座点的信号星座执行星座整形的方法。所述方法包括将某个成本与多个星座点中的各星座点相关联；定义块的分层结构，所述分层结构具有包括至少第一和最后一层的多个层；各层具有的块比其前一层少；其中第一层是通过如下步骤形成的：按照成本将所有星座点排序，然后将最低成本的星座点组分配给第一整形分割（shaping partition），如果划分包括多个整形分割，则继续将次低成本的星座点组分配给第二整形分割，并依此类推直到将最高成本的星座点组分配给最后一个整形分割，其中根据各整形分割中星座点的成本来为该整形分割分配成本，以及每个整形分割是第一层块；其中,其他每个层上的元素是通过合并前一层的两个块形成的，且其成本根据该前一层的两个块的成本来进行分配，每一层的块根据成本由一个所述元素或一组所述元素组成；最后一层只有一个包括多个元素的块；整形增益通过只映射到最后一层的元素的子集来取得。

在一些实施例中，分配给每个星座点的成本是其能量。

在一些实施例中，至少一些层块（layer block）只是前一层块的重新排序合并。

在一些实施例中，对于其中的块是前一层的块重新排序合并得到的块组的层，所有组都具有相同的大小。

在一些实施例中，在其中的块是前一层的块重新排序合并得到的块组的至少一层上，所述组具有不同的大小。

该方法还可以包括按如下步骤执行寻址：应用输入比特集合的第一子集来识别最高层的块中的元素；应用输入比特集合的后续子集来识别后续层上的块，其中在第一层识别特定的整形分割；在第一层，应用输入比特的最后一个子集来识别在第一层上识别的特定整形分割中的特定的信号星座点。

在一些实施例中，采用基于霍夫曼树的寻址方法。在其他实施例中，采用基于固定树的寻址方法。

在一些实施例中，采用了 256 个点星座；层 1 有 16 个分割，各包括 16 个星座点；层 2 有 8 个块，各包括 16 个元素；层 3 有 4 个块，各包括 8 个元素，

这 8 个元素的大小为{16、16、32、32、32、32、32、64}；层 4 有 2 个块，各包括 64 个元素；以及层 5 只有一个块，包括 4096 个元素。

在一些实施例中，输入比特包括多个数据比特和至少一个伪比特，以及该方法还包括对该至少一个伪比特的值的每种排列重复执行 16 的方法。

在一些实施例中，该方法对多个信号星座中的每一个重复执行，以生成每个信号星座的相应的多个整形输出。然后通过从每组相应的多个整形输出中适当地选择可使峰值均值功率降低的一个整形输出。

本发明的另一个宽的方面提供一种多谐波调制器，它包括：信号发生器，适于针对多个中心频率中的每一个中心频率生成包括多个星座点的相应信号星座的信号；星座映射器，适于针对每个谐波和 M 个数据元素值的每种排列，定义具有至少一个星座点的相应集合，其中任何一个星座点可用于表示 M 个数据元素值的排列，从而提供可选择用于表示 M 个数据元素值排列的星座点，至少一个具有至少一个星座点的所述集合具有一个以上的星座点；峰值平均功率降低功能（PAPR 功能），适于针对每个中心频率，从通过所述星座映射器映射到的至少一个星座点中选择一个星座点，以使所述多谐波信号的峰值对平均功率比降低；其中，所生成的信号利用该峰值平均功率降低功能选择的星座点发送。

本发明的另一个宽的方面提供一种调制器，它具有：整形装置，用于对多个信号星座中的每一个多次执行星座整形，以生成相应的多个整形输出；以及峰值平均功率降低装置，用于从每组多个整形输出中选择不同的单个整形输出，以使通过合并各单个整形输出形成的信号的峰值功率降低。

要注意，本发明的其他实施例提供一种计算机可读媒体和/或处理平台，例如适于实现上述或下文描述的方法/系统中的任何一项的通用目的处理器、定制的处理器、FPGA、DSP、ASIC。

附图简述

下文将参考附图详细说明本发明的优选实施例，附图中：

图 1 是说明采用常规多载波调制方法可如何导致高峰值功率的实例；

图 2 是说明采用本发明实施例提供的方法可如何将 36 QAM 用于降低峰值

功率的实例;

图 3 是选择使 PAPR 降低的星座点的第一方法的流程图;

图 4 是选择使 PAPR 降低的星座点的第二方法的流程图;

图 5 是采用星座整形（constellation shaping）和 PAPR 降低算法的 OFDM 发射机结构的方框图;

图 6 是图 5 的星座整形算法的简化方框图;

图 7a、图 7b 和图 7c 分解说明在图 5 的星座整形算法中可如何执行寻址操作;

图 8、图 9 和图 10 是说明采用各种星座整形算法的图 5 所示系统的性能结果的图示。

优选实施例的详细说明

根据本发明的实施例，在多载波系统中使用信号星座，其中采用了针对至少一些数据元素组合的冗余星座点。数据元素可以是比特，但并非必须。多载波系统可以是例如 OFDM 或其他形式的多谐波调制系统。

针对多载波的每个载波频率，定义具有多个星座点的信号星座。在每个符号持续期间，在多个载波的每个载波上发送星座点之一。

参考图 2，其中显示了一个有 N 个载波的实例，其中每个载波具有一个相关的 36-QAM 星座。第一个、第二个和第 N 个载波分别具有各自相关的 36-QAM 星座，一般地用标号 20、22 和 30 表示，第三个载波至第 N－1 个载波的细节则未显示。每个星座点代表在载波频率上发送的一对 I 和 Q 振幅分量。但是，对于每个载波，并不按每符号 $\log_2 36$ 个比特（约每符号 5.2 个比特）映射到这样的星座上，而是每符号映射数量减少的比特，在本实例中，每符号只映射两个比特。因为通常只需要四个星座点来表示两个比特，但有 36 个星座点可用，所以可以将 36 个可用星座点中的 9 个不同星座点分配用于表示各两比特组合。

这在图 2 的实例中用不同的阴影图案来表示。在第一个载波的星座 20 中，有九个带右斜条纹的星座点 30，它们分配来表示两个比特的四种可能排列的第一种排列；有九个带左斜条纹的星座点 32，它们分配来表示两个比特的四种可能排

列的第二种排列；有九个带网格填充的星座点34，它们分配来表示两个比特的四种可能排列的第三种排列；以及有九个无填充的星座点，它们分配来表示两个比特的四种可能排列的第四种排列。其他载波的星座22和30类似于图2所示的星座。九个星座点中的任何一个可用于表示相同的两个比特。

不必为每对比特定义相同数量的星座点，但优选实施例中却是如此。同时也不必如以上示例一样用星座点来表示等间隔的给定比特对，但优选这样做，因为这为降低峰值功率提供了最大的灵活性。当然应理解，可将任意星座用于N个载波中的每一个载波，它们甚至无需是同一个星座。而重要的是，每个星座包括一些冗余映射，更一般地说，至少一个星座需要包括一些冗余映射。冗余映射数量越多，则越容易降低峰值功率，如下文所述。

就图2的示例而言，多载波信号将由从与要在第一载波上传输的两个比特相关的第一载波的九个星座点组中选择的一个星座点，加上从第二载波的九个星座点组中选择的一个星座点等类推得到的星座点组成。因此有 9^N 种不同的方式来表示相同的2N比特序列。

再次参考图1，回顾一下，假定瞬间发射的多载波信号的峰值功率是通过对每个载波执行复振幅的复加法而确定的。对于给定的N比特序列，如果采用图2的星座，则用于表示相同N比特序列的 9^N 种不同的可能方式中的每一种会导致相应不同的多谐波信号复数加，从而会得到不同的峰值功率。选择一种得到可接受的且多数情况下降低了的峰值功率的特定表示。

在以图3的流程图说明的有可供选择的星座点的第一实施例中，说明了一种方法，此方法包括：对所有排列估算峰值与均值功率比（步骤3-1），然后搜索可供选择的星座点的所有可能排列，以得到产生最小可能的峰值对平均功率比的一组星座点（步骤3-2）。在上述实例中，这包括搜索所有所述的 9^N 种不同的可能的映射方式，以得到具有最小峰值功率的一种。在一个实现方案中，这种搜索步骤是在每次要调制一组给定的2N个比特时以实时方式执行的。在另一个实现方案中，假定答案总是相同，则如果2N足够小，就可以先验地对每个长度为2N的可能的比特序列进行搜索，以得到一个查找表，用以将每个2N比特序列映射到可能使峰值功率绝对最小的对应排列。

在另一个实施例中，并不搜索所有可能的映射方式，而是建立一个预定阈值来表示可接受的峰值功率。然后以实时方式对可能满足此阈值的任何可能的映射方式执行搜索。当找到具有小于阈值的峰值功率的可能的映射方式时，停止搜索。

任何适合的搜索技术都可以用于搜寻针对给定输入的、将导致可接受的峰值功率的适当的特定映射。图 4 的流程图说明一种例示搜索技术。在此示例中，搜索单星座点的组合从步骤 4-1 开始，在此步骤随机地从每组特定的不同的多个星座点中选择一个星座点。接下来在步骤 4-2，利用随机选择的星座点估算峰值对平均功率比。在步骤 4-3，如果峰值功率比小于预定阈值，则停止搜索。如果峰值功率比大于预定阈值，则在步骤 4-4 重复执行如下步骤直到峰值对平均功率比小于预定阈值为止：将随机选择的星座点中的至少一个星座点改为其所属的特定的不同星座点的集合中的另一个星座点，并重新估算峰值对平均功率比。此步骤重复执行，直到峰值对平均功率比低于阈值为止。

采用星座整形和峰值对平均功率比降低算法的 OFDM 发射机结构

下文将在 OFDM 发射机上下文中描述上述峰值对平均功率比降低算法的详细实例。此实施例的另一个特征还在于提供一种星座整形的创新方法。

本发明此实施例提供的 OFDM 发射机结构如图 5 所示。该发射机具有作为比特装入器 500 的输入的输入比特流 498，其中数据比特分成 J 组，一般地用 501 表示。假定有 J 个不同子载波组。在图 5 中，块 502 和 504 表示对 J 个不同子载波组中的两组子载波执行的星座整形算法，对其他子载波组也设有类似的算法块。各星座整形算法 502 和 504 快的输入由两个比特集组成，即由比特装入器 500 形成的 J 组比特 501 之一的数据比特以及（一般地用 506 表示）PAPR 降低算法生成的一个或多个伪比特。PAPR 降低算法 506 生成的伪比特一般地用 508 表示。PAPR 降低算法 506 确定这些伪比特的值，如下所述。

星座整形算法块 500 和 502 的输出确定各子载波组的二维（2-D）符号。进一步地详细说明本实例，其中，为 J 个比特组中的每组分配 16 个子载波，在此情况下共有 16×J 个子载波。对每个子载波组，单独使用星座整形算法。

就此实例而言，每个星座整形算法块对16个子载波中的每一个载波定义2-D符号。第一个星座整形算法块502的输出用503表示，最后一个星座整形算法块504的输出用505表示。所有星座整形算法块的输出都输入到IDFT（反离散傅立叶变换）功能块510。因此在本例中，对IDFT 510的输入是N=16×J维复矢量，其中N表示发射机中所用载波（谐波）的数量。每个载波从2-D QAM星座发送一个点。

调制由IDFT 510来执行。IDFT 511的输出通过一个并串转换器512，并对每个OFDM符号添加循环前缀以生成总输出。

采用星座整形算法将伪比特508和数据比特501映射到星座点，下文将对此予以说明。

星座整形算法

整形是一种用于使发送数据所需功率相对于未整形的（立方形）星座所需功率降低、同时使星座点之间的最小距离保持相同的方法。整形增益是通过采用比未整形星座更大的星座大小来实现的，其中星座大小的增加由星座扩展比（CER）给出。CER定义为整形星座的每2维符号的点数与在未整形星座中取得相同总速率所需的每2维符号最少点数之比。

在图5的示例结构中，针对每个32维矢量（16维的复维数）提供了独立的星座整形算法。每个载波发送一个2维点，因此一组16个2维载波表示32维矢量。每个这样的组将被称为“帧”，而每个2维点将被称为“符号”。

在图6中显示了一个非常简单的星座整形示意图。如上所述，每个星座被分为M个符号环或符号壳。一个壳编号定义这样一个壳，壳中的一点定义一个符号。例如，对于每个帧，可以利用壳映射器602将K个整形比特600映射到十六个壳$\{m_1、m_2...m_{16}\}$，而u个未编码的数据比特604利用信号点映射606选择各壳中的点。由图5的PAPR降低算法506输出产生的伪比特被用作K个整形比特600之一。假定B是每个整形帧发送的比特数，可以发现B=K+16u（因为每个帧有16个符号）。壳映射600的作用在于为16个子载波中的每一个载波选择16个壳之一。于是，对于这样选择的每个环，u个未编码比特选择壳内的一个

特定点。

假定 M 是每符号壳数量，换言之为星座所划分成的壳数量。于是，一个符号所用的星座点数量 L 则由如下公式给出：$L = M \times 2^U$。可以确定 CER，（每符号实际星座点数）/（每符号最小星座点数）。因为每个符号承载 u 个数据比特 + K/16 个整形比特，所以最小的星座大小为 $2^{u+K/16}$。实际星座大小是 $M \times 2^u$。因此 CER 为 $\frac{Mx2^u}{2^{u+K/16}} = \frac{M}{2^{K/16}}$。

在优选实施例中，选择 M 应使 2 维 CER 至少为 1.35，即 $M \geq 1.35 \times 2^{K/16}$。

举例说明，对应于各种 K 值，为达到 1.35 的 CER 所需的最小 M 值如表 1 所示。

K	$M \geq 1.35 * 2^{K/16}$
25	M≥3.99
41	M ≥ 7.97
57	M ≥ 15.95

表 1 不同的 k 和 M 值

符号的 L 个星座点按如下排序：壳 0 包括 2^U 个最小能量点、壳 1 包括 2^U 个次最小能量点以及依次类推直到壳 M-1。为每个壳 i 分配成本 c_i，成本 c_i 与壳 i 中点的平均能量成比例。

各种将 K 个整形比特映射到环（图 6 的壳映射 602）的树结构图 602 均可以采用。接下来，说明另外两个可以采用的树结构实例，一个是基于霍夫曼树的，而另一个是基于固定树的。

霍夫曼树

下文给出在每个子载波采用一个 QAM－64 星座的上下文中的一个实例。映射执行以“层 1”为底部而以“层 5”为顶部。参考图 7A，层 1 总体用标号 700

来表示，它由 16 个块 702 组成，其中每个块表示一个发送的符号，一个符号是来自 QAM-64 星座的一个点。块标记为 QAM64-1（层_1-1）、...、QAM64-16（层_1-16）。可用于每个 2 维符号的 64 个点按其能量非降序排序，并编组成 16 个环，每个环 4 个点。（环 0 由 4 个能量最低的点组成，而环 15 由 4 个能量最大的点组成）。环的能量是该环中 4 个点的平均能量。层 1 上给定块的每个环是 Q_i，i = 0、...、15。

层 2（总体用标号 704 表示）由八个块 706 构成。两个相邻层 1 上的块合并成一个层 2 上的块。层 2 上的每个元素具有指向层 1 上不同块中的两个环的指针。换言之，层 2 上的每个元素对一定 i 和 j 为 Q_iQ_j。如同层 1 一样，将层 2 上的每个块的元素组成 16 个环。因为层 1 上的环有 256 种不同的可能组合，所以每个层 2 块具有 256 个元素。层 2 上每个元素的能量是层 1 上层 2 元素所指向的那两个环的平均能量。同样，这 256 个元素根据它们的能量排序，并编组成各具有 16 个元素的 16 个壳。（壳 0 由 16 个具有最低能量的点组成，壳 15 由 16 个具有最大能量的点组成）。这样就为各包含十六个 Q_iQ_j 组合的每个层 2 块定义了十六个壳 Si，i=0、...、15。

层 3（总体用标号 710 表示）由 4 个块 712 组成。两个相邻层 2 块 706 合并成一个层 3 块 712。层 3 块中的每个元素具有指向层 2 上不同块中的两个壳的指针。换言之，层 3 上的元素对一定 i 和 j 为 S_iS_j。层 2 上的壳有 256 种不同的可能组合，所以每个层 3 块具有 256 个元素。这 256 个元素根据它们的能量排序，并编组成 8 个壳 T_i，i = 1，8。为了降低寻址的复杂性，此阶段的合并操作利用数量不等的元素来实现。一个具有良好性能的特定实例是，选择大小为{16、16、32、32、32、32、32、64}的壳。具有最小能量的元素置于较小的壳中。壳 0 由 16 个具有最低能量的点组成，壳 7 由 64 个具有最大能量的点组成。这样就为每个层 3 块定义了 8 个壳 T_i，i=0、8，每个层 3 块包含数量可变的 S_iS_j 个组合。

总体用标号 714 表示的层 4 由两个块 716 组成，其中两个相邻的层 3 块 712 合并成一个层 4 块 716。层 4 块中的每个元素具有指向层 3 上的不同块中的两个壳的指针。因为在层 3 每个块具有 8 个壳，层 3 上的壳有 64 种不同的可能组合，所以每个层 4 块具有 64 个元素 T_iT_j(对于任意 i 和 j)。元素按它们的能量排序，

但在层 4 不分组。因此，层 4 具有 64 个按能量排序的元素 U_i。

总体用标号 718 表示的层 5 由一个块 720 组成，该块由两个层 4 块 718 组成。层 5 块中的每个元素具有指向层 4 上的不同块中的两个元素的指针。因为在层 4 每个块具有 64 个元素，所以层 4 上的元素有 4096 种不同的可能组合 U_iU_j（i 和 j 任意），因此层 5 块具有 4096 个元素。层 5 元素根据它们的能量排序而不进行任何分组，得到一个层 5 元素 V_i(i = 1、...、4096)的集合。

在没有整形的情况下，在层 5 选择一个元素需要 12 比特。整形增益是通过将少于 12 的比特映射到层 5，以使映射到给定比特集合的点有冗余来实现的。层 5 上的子集按最小能量来选择。

在由一组 R 个比特组成的输入中，其前 x 个比特在层 5 用来选择元素。需要 12 个比特来选择一个元素，但是为了实现整形增益，要采用少于 12 的比特，最好用 7、8 或 9 个比特来选择层 5 上的元素。

层 5 上的每个元素指向层 4 上的两个元素，而层 4 上的每个元素指向层 3 上的两个壳。因此在层 5 选择的元素指向 4 个层 3 上的壳。在层 4 没有进行分组，所以不需要额外的比特来选择层 4 上的元素。选择层 3 上的每个壳中的元素以下至层 2。因为每个壳中的元素数量不相同，所以在层 3 需要从输入中读取数量可变的比特。为了从大小为 16、32 或 64 的壳中选择元素，分别需要 4、5 或 6 个比特。因此，从输入读取 4、5 或 6 个比特，以在层 3 选择每个块的壳中的元素（根据壳大小）。

在层 3 选择的每个元素指向层 2 上的两个壳，其中每个壳具有 16 个元素。为下至层 1，需要选择层 2 每个壳中的元素。对于层 2 上的每个块，从输入读取 4 个比特，以选择壳中的 16 个元素之一。在层 2 选择的元素指向层 1 上的两个环，其中每个环具有 4 个点。从输入读取 2 个比特，以选择层 1 上的环中的 4 个点之一。

假定 R_n 是从层 n 下至层 n-1 所需从输入读取的比特数量，n=1 至 5。在层 1，对于每个块，要从一组 4 个点中选择一个点，即每个块需要 2 个比特。因为有 16 个层 1 块，所以 $R_1=16\times2=32$。在层 2，对于每个块，要从一组 16 个元素中选择一个元素，所以每个块需要 4 个比特。因为有 8 个层 2 块，所以 $R_2=8\times4=32$。

层3所需的比特数量有所不同，因为该层的壳大小是不相同的。R_3可以在16（对应于所有层3块中具有16个元素的每个壳）和24（对应于所有层3块中具有64个元素的每个壳）之间变动。R_4为0，因为在层4不进行分组。R_5选择按如下要求变化，即使寻址中所用的比特数量为固定的，且R_n的和（n=1至5）为输入的大小R。

假定层5具有2^x个元素，这样在层5使用x个比特来选择一个元素。由此得到指向两个层4元素的指针，注意在层4不需要任何比特。从层4往下，由指针提供对壳的引用，所以需要一定数量的比特来选择该给定壳内的元素。所需的比特取决于壳的大小。因为层3上的壳大小各不相同，所以该层所需的比特不是常量，因此每个壳必须指定所需比特，并相应地从输入中捕获它们。层2和1具有16的固定壳大小，因此需要4个比特用于选择。如下条件是必需的：信号星座所用的总比特数应该是常数值，又因为层3所需的比特有所不同，所以必须引入一种方法来调整在顶层选择所用比特，以便比特的总和始终为常数。

假定在顶层，可能有总计I（= 4096）个元素，需要大小为i的子集来容纳总数固定为R的输入比特。这表示只采用了I个元素中的i个。i值的选择是通过试验来确定的，即对不同的值进行尝试，直到R=输入的大小、K=整形比特的数量都为整数为止。正如以下将要说明的那样，121个输入比特中的57个比特用作整形比特，需要4096个元素中的i=194个元素。请注意R=K+16u，因此u=4意味着用四个信息比特来从层1壳中16个元素选择点。

为了确定适当的i，K按如下公式计算：

$$K = \log_2\left[\sum_{n=1}^{i} 2^{R'_n}\right]$$

其中R'_n是从顶层元素n沿树形结构向下寻址所需的整形比特数量，而$K - R'_n$则是在顶层用于选择元素n的比特数量。

图7C是用于寻址的示例霍夫曼树结构，图7B则显示用于寻址的树层次结构。图7C显示4096个可能的元素中的总计194个元素。前70个用于层5上的7位寻址。其余58个7位地址中的54个用作8位地址的前7位，即有$54 \times 2 = 108$个可能的8位层5地址。58个7位地址的其余4个用作9位地址的前7位，即有

$4\times4=16$ 个可能的 16 位层 5 地址。因此，我们有 $R_n^{'}=48$、49、50 和(70×2^(50)) + (108×2^(49)) + (16×2^(48))=2^(57) = 2^K=可能的整形映射数量。

如上所述，计算不同的 i 值，直到 R 和 K 为整数为止。再者，应该选择 R 和 K 以提供所需的 CER 值，即：

$$\frac{256}{2^{R/16}}\cong 1.35$$

假定 $b = K\text{-}R_n^{'}$，则 $b\in\{b_1,\ldots,b_j=b_1+j-1\}$，其中 b_1 是 b 的最小值，j 是 b 可取的不同值的数量。子集中的顶层元素根据 b 的值分组。然后利用霍夫曼树结构为每个元素分配代码。将具有从 b_1 至 b_j 的值的元素从左到右置于树的底部的不同层上。将每个元素以精确的高度 b 放置于树中，这意味着该元素可以用长度为 b 的代码来表示。这等效于为具有值 b1 的元素分配来自从 0 到 s_1-1 的第一组元素的代码，s_1、s_2 ... s_j 分别是值各为 b_1、…、b_j 的各个集合中的元素的总数。对于具有值 b_2 的元素，添加一个最小位（significant bit），这样，代码从 $2\times s_1$ 开始到 $(2\times s_1+s_2-1)$。同样地，具有值 b_3 的元素从 $2\times(2\times s_1+s_2)$ 开始到 $[2\times(2\times s_1+s_2)]+S_3-1$，并依此类推。

层 3 壳的优化配置之一是{16、16、32、32、32、32、32、64}（参考图 7C）。已发现，为发送 R=121 个比特，i 应为 194。在 i 个元素的子集内，只有 j=3 个不同的 b 值，它们是 7、8 或 9。因此，在层 5，将 194 个元素分成 3 个集合，且可以通过算法操作定位指定的元素。在层 4 有两个各包括 64 个元素的块。最后，层 3 至层 1 上的所有块各有 256 个元素，只是壳分组方式有所不同。

因此，在层 5，只选择映射输入所需的 4096 个元素的子集。在本例中，因为将在层 5 使用 7、8 或 9 个比特，所以只需要 4096 元素的前 194 个元素，它们是层 5 中具有最小能量的 194 个元素。这 194 个元素是根据层 3 所需比特数量确定的。这 194 个元素由三组组成，即分别具有 18、17 和 16 个比特的 R3 值、大小分别为 S_1=70，S_2=108 和 S_3=16 的组 1、组 2 和组 3。因此，R_5 对于组 1、组 2 或组 3 必须分别为 7、6 或 9，以使 R_3 和 R_5 中总计使用 25 个比特。

然后将霍夫曼树结构用于对层 3、4 和 5 寻址。从输入中读取 7 位。如果 7 位的值小于 70，则选择组 1 中的元素。由于组 1 中的所有元素在层 3 需要 18 个

比特，所以满足 $R_3+R_5=25$ 个比特。

如果 7 位的值大于或等于 70，则从输入再提取一个比特，并将 8 位的值与 248 比较。如果 8 位的值小于 248，则选择组 2 中的元素。否则，从输入又提取一个比特，并根据 9 位的值选择组 3 的元素。结果，本算法以将固定数量的输入比特映射到 16 个环以形成一帧而结束。

为了作进一步的说明，以下将以不同方式来说明上述实例。假定 2 维子空间中有 16 个壳，则 4 维子空间中 2 维子星座的笛卡儿积中有 16×16=256 个元素。4 维元素根据它们的平均能量排序，16 个后续元素合并成一个 4 维壳。由此得到在 4 维子空间中基数相等的 16 个壳。同样地，4 维壳的笛卡儿积（在每个 8 维子空间中得到 256 个元素）根据它们的平均能量来排序，将后续元素合并到 8 维壳中。为了降低寻址的复杂性，此阶段的合并操作是利用后续 8 维壳中不等数量的元素来实现的，具体来说，合并到后续 8 维壳中的元素（按平均能量递增排序）的数量分别等于{16、16、32、32、32、32、32、64}。这得到 8 维子空间中的 8 个壳。接着，我们得到 16 维子空间中的 64 个元素（未合并）和 32 维空间中的 4096 个元素（未合并）。注意，16 维和 32 维元素的基数不相等（这些基数全部是 2 的整数幂）。最终的星座从具有最低能量的 32 维元素中选择，以使总比特率为 57。注意在未整形的情况下，比特率将是 16×4 = 64。这意味着整形冗余度等于 7 比特/32 维，因此得到：

$$CER = 16/(2^{57/16}) - 1.35$$

在分层结构的最高层上寻址是利用如图 7B 和 7C 所示的霍夫曼树来实现的。第一步是确定所选 32 维元素属于组 I、组 II 还是组 III。这是通过假定将“0”分配给该树每个节点的左分支，将“1”分配给右分支而获得的二进制数来标记 194 个元素来实现的。然后，通过将每个最终节点（对应于一个元素）的分支的二进制标号级联而得到其标号（其中最高有效位对应较接近树顶层的值）。这将使最终节点的标号按从左到右递增排序。注意，最终节点的标号由 7 比特（对应组 I）、8 比特（对应组 II）以及 9 比特（对应 III）组成。在此情况下，要选择 32 维元素，我们就从输入流中提取 7 比特，并将其数值与阈值 T1 比较（即组 I 中最后一个元素的标号）。如果该标号小于或等于 T1，则属于组 I，于是还有 57-7 = 50

个比特要继续在所选元素内进行寻址处理。否则我们从输入流中再提取一个比特，得到 8 位标号。我们将所得到的 8 位标号的数值与阈值 T2（即组 II 中最后一个节点的标号）比较。同样地，如果该标号小于或等于阈值 T2，则我们利用剩余的比特在组 II 内继续进行寻址处理（在本例中，剩余 49 个比特）。如果该标号大于 T2，则我们从输入流再提取一个比特，并利用得到的 9 位在组 III 内选择元素（在本例中，剩余 48 个比特可供选择组 III 内的元素）。

在（如上所述）选择 32 维中 194 个元素之一后，继续在所选元素内寻址，提供了设有 194 个存储位置且各存储位置为 $4\times3 = 12$ 位的查找表，其中 3 位地址段指向构建给定的 32 维元素的 8 维壳（注意每个 32 维元素只是四个 8 维壳的笛卡儿积）。这样得到 $194\times12=2328$ 位的 ROM。

为了继续在 8 维壳中寻址，注意到所选的 8 维壳由 16，32 或 64 个 3 维元素（分别对应于 4、5、6 个比特）组成。在此情况下，我们（为每个 8 维子空间）从输入提取另外 4、3 或 2 个比特，以将每 8 维的比特总数增加为 8 个比特。我们使用这些 8 位地址选择每个 8 维子空间中的一个 8 维元素。为此，提供了设有 256 个存储位置且各存储位置存储位置为 $4\times2 = 8$ 位的查找表，其中这些 8 位地址的每个 4 位地址段指向构造给定 8 维元素的 4 维壳。这些位用于选择每个 4 维子空间中的一个 4 维壳。

为了继续在 4 维壳内寻址，提供了设有 256 个存储位置且各存储位置存储位置为 $4\times2=8$ 位的查找表，其中这些 8 位地址的每个 4 位地址段指向组成给定 4 维元素的 2 维壳。这些位用于选择每个 2 维子空间中的一个 2 维壳。如上所述，（根据每个 2 维壳内的信号点的数量）将采用另一组输入比特来选择每个 2 维子星座内的最终点。这样最终结束寻址。

所建议的寻址方案的总存储需求等于:

$M_{总计} = 2328 + 2048 + 2048 = 6424$ 个比特 ~ 0.73 K

注意除非两个作为组成部分的较低维分量相同，如 $A\times A$，否则笛卡儿积中的元素始终以（能量相等的）对出现，即 $A\times B$ 和 $B\times A$。此特性可用于使所需存储空间大小减少约 2 倍，而代价是非常少的比较次数（将标号与具有相同分量的元素的标号所对应的某些锚点作比较）。仿真结果表明所建议的星座达到约 1

分贝的整形增益。

作为一个特定实例，假定每个 2 维子星座由划分成 16 个壳、各壳含 16 个点的 256 个点组成。输入由 57 个整形比特加另外的 64 个比特组成。这 57 个整形比特在所提出的整个寻址方案过程中参与 2 维壳的选择，并将用于在每个 2 维子星座中选择一个唯一的 2 维壳。其余 64 个比特将分成 16 个部分，各部分具有 4 个比特，这 4 个比特中的每一个将在随后用于从所选 2 维壳的每一个 2 维壳中可选的点中选择唯一的一个点。我们只需要关注利用 57 个整形比特选择 2 维壳的操作。在这些 2 维壳内选择最终的 2 维点属于次要任务。此过程同样可应用于任何其他 2 维点的数量非 256、而 2 维壳的数量为 16 的实例情况。例如，如果 2 维子星座由 64 个点组成，则有 16 个 2 维壳（各具有 4 个点）以及输入比特总数将是 57 比特（用于整形）加用于选择最终 2 维点的 32 个比特（所述 57 个整形比特的寻址过程对缩放 2 维子星座大小的所有情况都完全相同）。下文中，我们假定 2 维子星座由 256 个点组成。有 57+64=121 个输入比特，我们选择其中的前 57 比特用作寻址（整形比特）。我们现在来考虑 57 个整形比特的如下情况：

情况 I: 0010010....

前 7 位是 0010010，其值为 18（小于 T1=69）。在此情况下，将在图 7C 所示的组 I 中进行选择。值 18 将指向相应查找表中的某个位置，该查找表包含 4 个各为 3 位的指针，用于指定相关的 8 维元素（此类 8 维元素有 4 个）。在知道这些 8 维元素之后，通过从剩余输入比特中顺序提取适当数量的比特并利用这些比特查询用于在 4 维和 2 维子空间内定址的相关查找表，以选择 4 维和 2 维内它们的相关元素，这样继续在那些 8 维元素内的每一个 8 维元素内寻址。

情况 II: 1010010 0 100100

前 7 位是 1010010，其值等于 82 且大于 T1=69。继续处理，我们从输入中再提取一个比特，它等于 0。然后将前 8 位，即值为 164 的 10100100T2=247 作比较。因为该标号小于 T2，所以在组 II 内进行选择。这时，还剩下 49 个整形比特可用于继续选择 8 维元素。

情况 III: 1111011 1 100100

前 7 位是 1111011，其值为 123，大于 69，而前 8 位即 11110111 的值是 247，所以在组 II 内进行选择，并将此前 8 位用于在组 II 中寻址。

情况 IV: 1111110 1 100100

前 7 位的值大于 T1=69，且前 8 位的值大于 T2=247，所以在组 III 内进行选择，因此提取前 9 位用于在组 III 内寻址。

固定树

下文将描述固定树实现方案的实例。固定树指以统一方式（将相等数量的元素合并成壳）来实现分层结构中所有层上元素的合并。这样就导致搜索相应的存储器查找表需要较大的存储空间、较少的计算次数。就层 1 和层 2 而言，固定树的结构与霍夫曼树相同。层 3 由 4 个块组成。两个相邻层 2 块合并成一个层 3 块。层 3 块中的每个元素具有指向层 2 上不同块中的两个壳的指针。与在层 2 一样，每个块具有 16 个壳，层 2 上的壳具有 256 种不同的可能组合，所以每个层 3 块具有 256 个元素。这 256 个元素按照它们的能量排序，并划分成 16 个壳，各壳有 16 个元素。

层 4 由 2 个块组成，其中两个相邻层 3 块合并成一个层 4 块。层 4 块的每个元素具有指向层 3 上的不同块中的两个壳的指针。与在层 3 一样，每个块具有 16 个壳，层 3 上的壳具有 256 种不同的可能组合，所以每个层 4 块具有 256 个元素。这 256 个元素按照它们的能量排序，并划分成 128 个壳，每个壳具有 2 个元素。

层 5 由 1 个块组成，该块由两个层 4 块组成。层 5 块的每个元素具有指向层 4 上的不同块中的两个元素的指针。与在层 4 一样，每个块具有 128 个壳，层 4 上的壳具有 16384 种不同的可能组合，所以层 5 块具有 16384 个元素。在层 5，元素根据它们的能量排序，但不进行任何分组。

89 个比特的输入中，前 7 位用于选择层 5 上的元素。因为 7 个比特可以选择 128 个不同的元素，所以在层 5，只使用前 128 个元素。层 5 上的每个元素指向层 4 上的两个壳。在层 4，需要在每个壳中选择元素以下至层 3。对于层 4 上

的每个块，从输入读取一个比特，以选择某个壳中的两个元素之一。在层 4 所选的元素指向层 3 上的两个壳，其中每个壳具有 16 个元素。对于层 3 上的每个元素，从输入读取 4 个比特来选择层 3 上的 16 个元素之一。在层 3 所选的元素指向层 2 上的两个壳，其中每个壳具有 16 个元素。同样地，对于层 2 上的每个块，从输入读取 4 个比特来选择层 2 上的 16 个元素之一。在层 2 所选的元素指向层 1 上的两个环，其中每个环具有 4 个元素。对于层 1 上的每个块，从输入读取 2 个比特来选择层 1 上某个环中 4 个点之一。

各层从输入读取的比特总数如下：在层 5 读取 7 个比特；在层 4 读取 2×1=2 个比特；在层 3 读取 4×4=16 个比特；在层 2 读取 8×4=32 个比特；以及在层 1 读取 16×2=32 个比特，总计 89 个比特。

也可采用不同的固定树结构。表 2 显示了另一种固定树结构，该固定树结构在层 1 具有 4 个环。

	固定树结构 1			固定树结构 2		
	壳数量	每个壳中点的数量	读取的比特数	壳数量	每个壳中点的数量	读取的比特数
层 1	16	4	2×16=32	4	16	4×16=64
层 2	16	16	4×8=32	16	1	0×8=0
层 3	16	16	4×4=16	16	16	4×4=16
层 4	128	2	1×2=2	12 8	2	1×2=2
层 5	16384	1	7×1=7	16384	1	7×1=7

表 2 两种可能的固定树结构

PAPR 降低算法

现在可以调整先前描述的 PAPR 算法，以适用于上述星座整形算法。再次参考图 5，每个星座整形算法 502 和 504 的输出由是 B 个比特生成的 32 维矢量。

在本实施例中，这 B 个比特由从输入比特流提取的 B-1 个数据比特和由 PAPR 算法 5O6 选择的一个伪比特 508 组成。当伪比特是 0 或 1 时，由星座整形算法 502 和 504 执行两次 B 比特寻址（addressing of B bits），分别对应于伪比特等于 0 或 1 的情况。这样每个 32 维矢量就有两种选择。对于采用 N 个谐波的 OFDM 系统，调制器的输入是通过使 J 个 32 维矢量级联而生成的，使得 J×16=N，每个载波发送一个 2 维点。对于 J 个 32 维矢量中的每一个 32 维矢量，我们可以从一组 2 个 32 维矢量中选择一个 32 维矢量，这样，根据伪比特的值可以以 2^J 种不同方式生成 2N 维矢量。因此，此上下文中 PAPR 降低算法 506 的目的是，从一组 2^J 个矢量中选择 2N 维矢量，此 2N 维矢量在调制之后产生最低峰值振幅。一般来说，2^J 是一个使穷举搜索因计算高度复杂而无法执行的大数值。

所述每星座两种选择与参考图 2 讨论的所述每星座多种（9 种）选择相似。在本例中，采用一个伪比特就得到每星座两种选择。更一般地说，可以采用较大数量的伪比特以得到每星座更多种选择。但是，为找到具有最低峰值功率的特定排列需要进行更多的搜索。

在一个实施例中，采用迭代搜索算法在 32 维矢量的两种选择之间进行选择。最初设置每个 32 维矢量的伪比特的值，使 2N 维矢量具有最低能量。注意，最低能量特性并不意味着经过调制后就会得到最低峰值振幅。接着，从第一 32 维矢量开始，使伪比特的值反转（即选择第一个 32 维矢量的另一种选择），同时保持其他伪比特的值不变。然后执行调制，并计算峰值振幅。如果在峰值振幅有所下降，则保留伪比特的新值；否则舍弃该新值。对所有 J 个 32 维矢量重复此过程，如此迭代多次。

该迭代搜索算法不是最优的，因为无法保证得到的结果给出全局最小的峰值振幅。一般从不同初始条件开始或以不同方式执行搜索算法将得到不同的结果。一种提高算法性能的方法，是从相同的初始条件开始但更改搜索模式来获得多个解。第一解可以通过从第一个 32 维矢量开始并更改第一、第二、第三等 32 维矢量的伪比特来获得。另一个解则可以通过从第 J-1 个 32 维矢量开始，并以降序方式更改 32 维矢量的伪比特，即顺次更改第 J-1、第 J-2 等 32 维矢量的伪比特而获得。可以采用不同的搜索模式，不同的搜索模式得到不同的解。最后，从所

获得的解中选择给出最小峰值振幅的解。

结果

已经获得 N=1024 个载波（其中每个载波发送一个 64 QAM 星座点）的仿真结果。仿真结果已通过发送 10^5 个 OFDM 符号得到。对于每个符号，存储具有最大峰值振幅和 OFDM 符号功率的样本。通过峰值振幅除以所有 OFDM 符号的平均功率而获得峰值对平均功率比。通过计算 PAPR 大于给定 PAPR 的符号的出现次数来绘制 PAPR 结果。

对于星座整形算法，每 16 个载波发送一个伪比特和 88 个数据比特，即 R=88+1=89。描绘了与霍夫曼树和固定树结构对应的结果。对于固定树，采用了表 2 中提供的两种结构。在图 8 中，描绘了采用和不采用星座整形算法时的 PAPR。迭代次数设为 12，只得到一个解。在图 9 中，迭代的次数设为 3，在该迭代搜索算法中得到 4 个解，并采用这 4 个解中最优的解。

对于获得的结果，如果对应某些伪比特组合的 OFDM 符号的 PAPR 小于 7 分贝，则不执行额外的迭代以及搜索另一个解。迭代搜索算法中所用 IDFT 的次数由（解的个数）×（迭代次数）得到，因此其结果与图 8 和图 9 所绘制的结果一样。

从这些附图中，可以观察到，星座整形算法中所用的树结构对结果有轻微影响。利用迭代搜索算法，得到一个解的仿真在符号削波概率为 10^{-4} 下获得了 7.9 分贝的 PAPR，而得到四个解的仿真在 10^{-4} 获得了 7.4 分贝的 PAPR。因此增加所得解的数量在一定程度上提高 PAPR 算法性能。

还取得了整形算法另一种版本（将 8 个而非 16 载波成组）的仿真结果。采用具有 4 个层次的固定树结构，表 5 中显示了该树结构。层 1、2、3 和 4 分别包括 8、4、2 和 1 个块。

	固定树结构 3		
	壳数量	每个壳中点的数量	读取的比特数
层 1	16	4	2
层 2	16	16	4
层 3	128	2	1
层 4	16384	1	11

表 3 固定数结构 3

在此情况下，通过 8 个载波发送的比特总数是 45，即每载波 5.625 个比特，CER=$16/2^{K/8}$=1.30。

图 10 中给出了不同树结构的结果。固定树结构 3 在符号削波概率（symbol clipping probability）为 10^{-4} 下将 PAPR 提高 0.4 分贝。已证明采用此星座整形算法的 OFDM 符号的平均功率比未采用整形算法的 OFDM 符号少 2.17 分贝。

所述算法的吞吐量比将 16 个载波成组的算法的性能略高（5.625 比特/每载波对 5.5625 比特/每载波）。因为每符号伪比特数比将 16 个载波成组的树结构多两倍，所以所用 IDFT 的数量是迭代搜索算法中的两倍。将比特寻址到环的复杂性较低，因为树结构较简单。

根据以上论述，对本发明作各种修改和变化是可能地。因此应理解，在所附权利要求书的范围内，可以以不同于本说明书中所描述的方式来实施。

参考文献:

[1] G. D. Forney，Jr.and L. P. Wei 的 “Multidimensional constellations--Part I; Introduction，figures of merit，and generalized cross constellations”，IEEE J. Select. Areas Coitimun.，vol. SAC--7，pp. 877—892，August 1989。

[2] A，K. Khandani and P. Kabal 的 “Shaping multi-ctim&nsional signal

spaces-Part I: Optimum shaping, shell mapping" , IEEE Trans. Inform. Theory, vol.IT-39, pp.1799-1808, Nov. 1993。

[3] A. K. Khandani and P. Kabal, "Shaping multi-dimensional signal spaces-Part II: shell-addressed constellations" IEEE Trans. Inform. Theory, vol.IT-39, pp.1809-1819, Nov.1993。

[4] F. R. Kschischang and S. pasupathy, "Optimal shaping properties of the truncated polydisc" IEEE Trans. Inform. Theory, vol.40, pp. 892--903, May 1994.

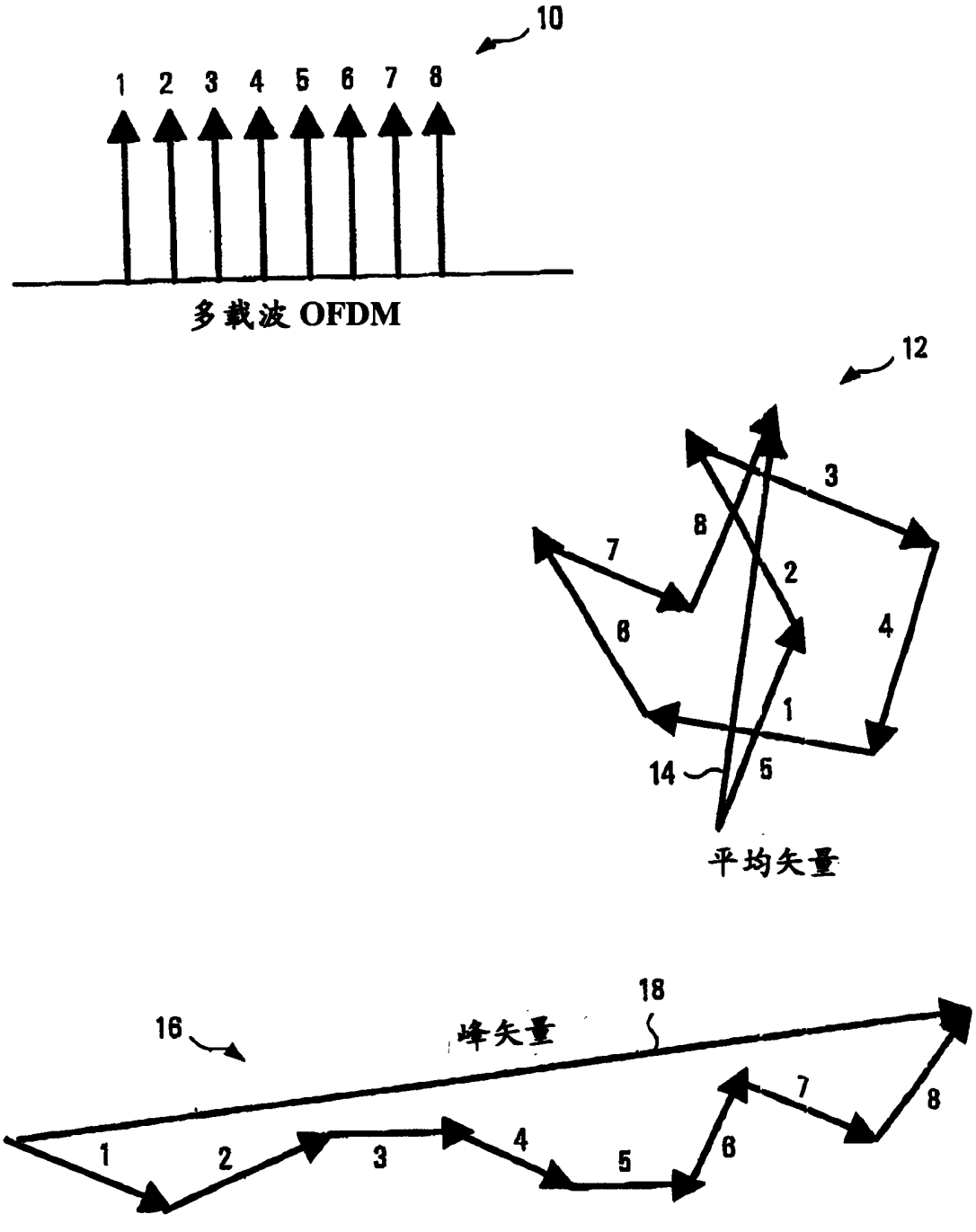


图 1

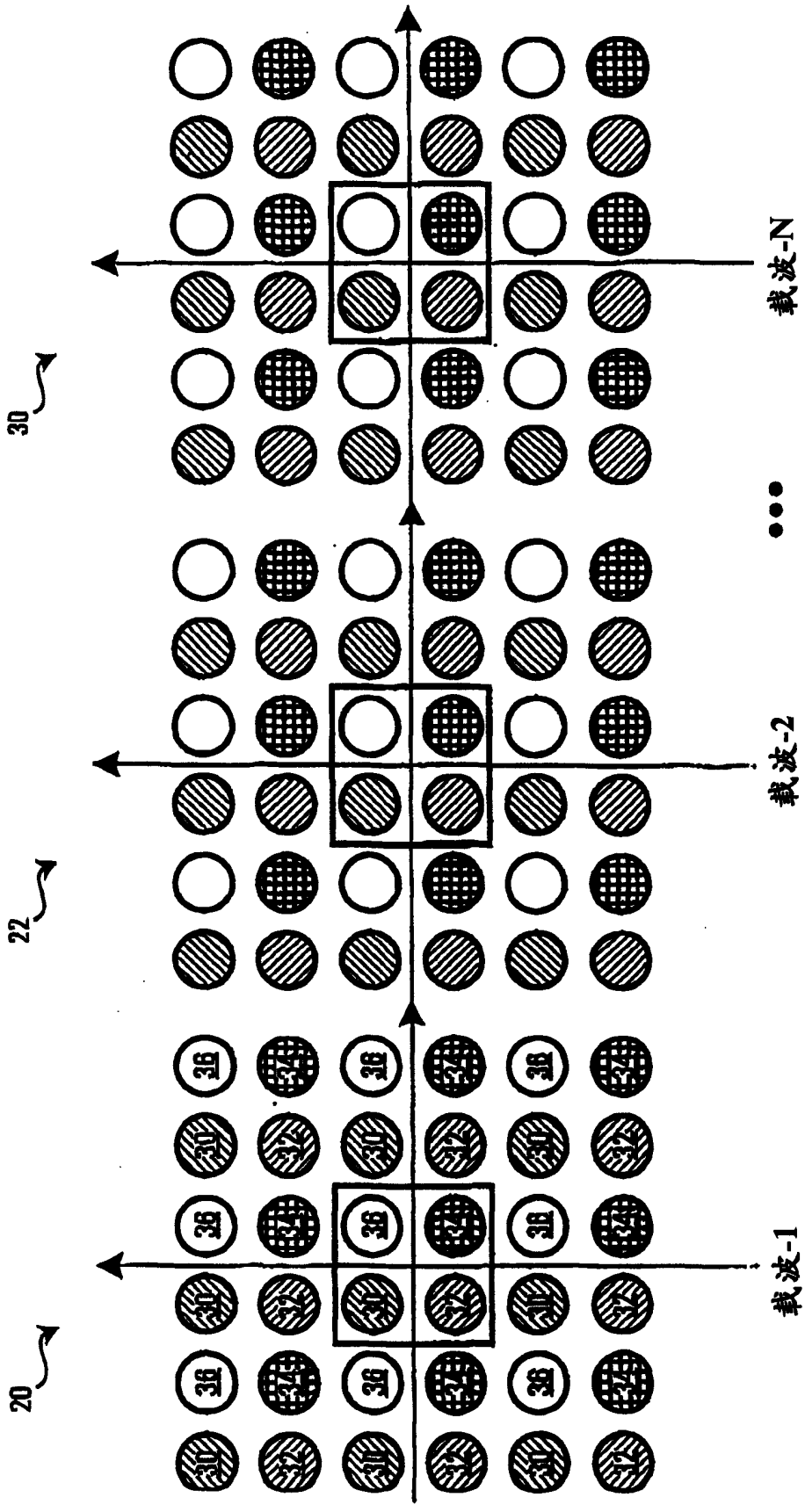
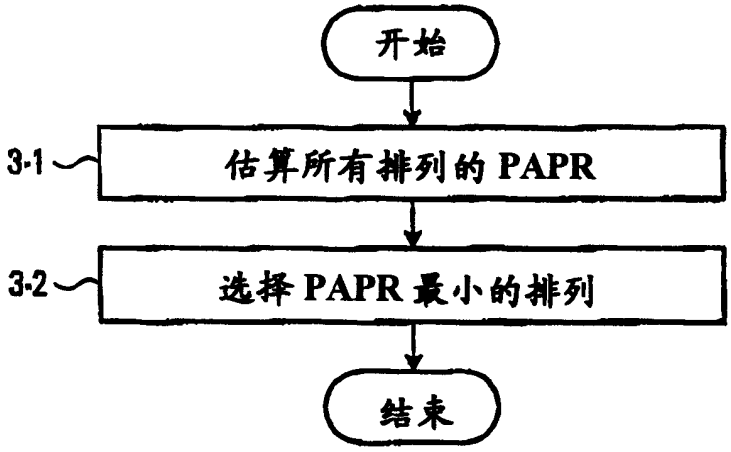


图 2

图 3


开始
3-1
估算所有排列的 PAPR
3-2
选择 PAPR 最小的排列
结束

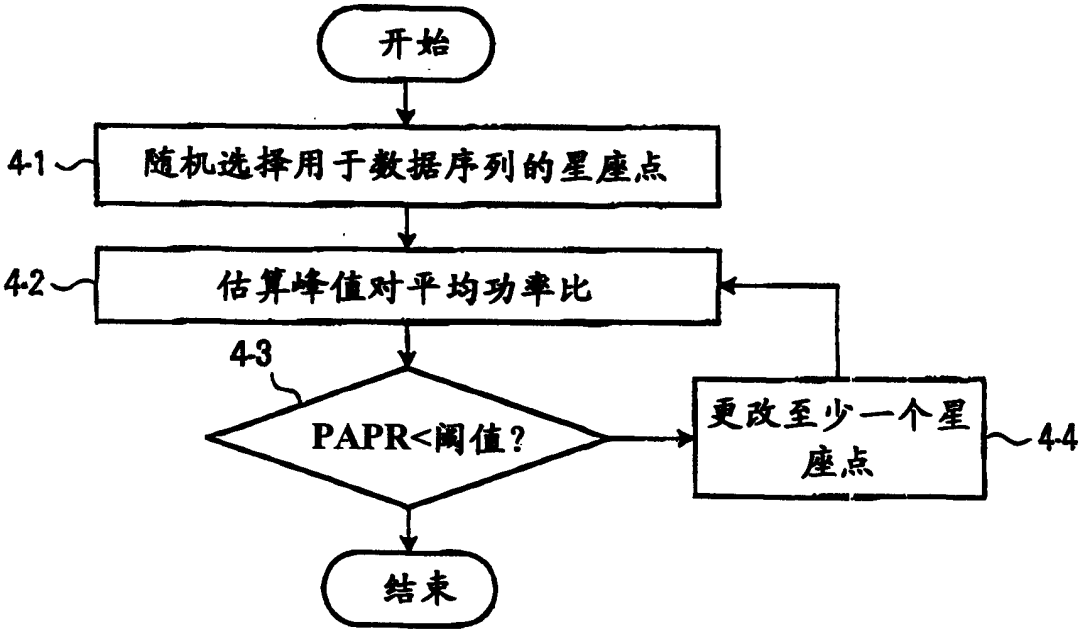

开始
4-1
随机选择用于数据序列的星座点
4-2
估算峰值对平均功率比
4-3
PAPR<阈值?
更改至少一个星座点
4-4
结束

图 4

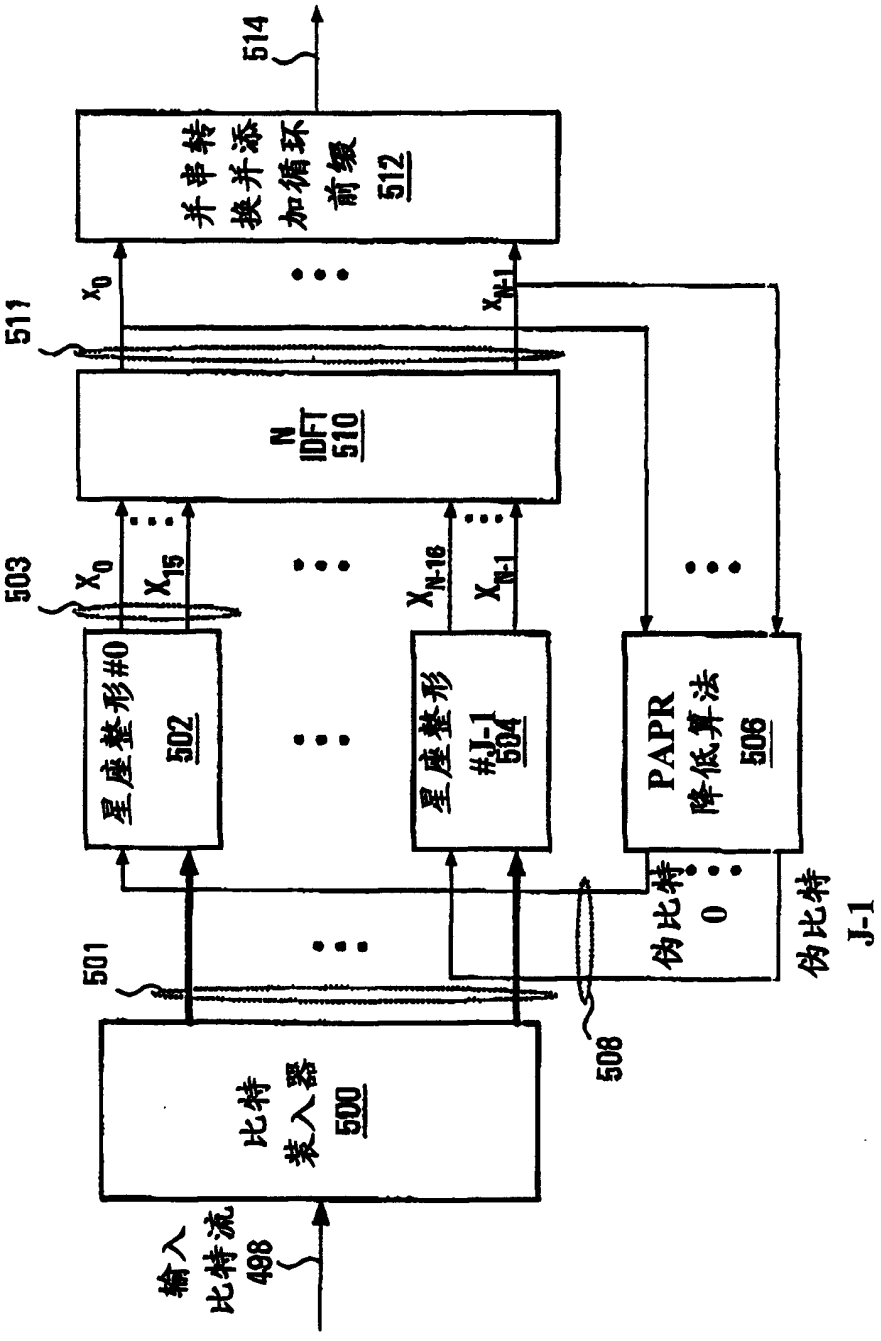

输入
比特流
498
比特
装入器
500
501
星座整形#0
502
503
星座整形
#J-1
504
PAPR
降低算法
506
伪比特
0
伪比特
J-1
508
N
IDFT
510
511
并串转
换并添
加循环
前缀
512
514

图 5

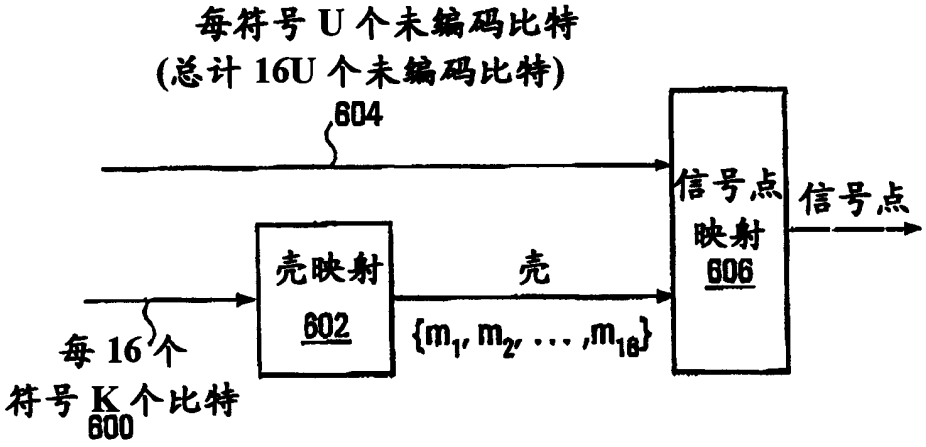

每符号 U 个未编码比特
(总计 16U 个未编码比特)
604
信号点
映射
606
信号点
壳映射
602
壳
{m1, m2, ... ,m16}
每 16 个
符号 K 个比特
600

图 6

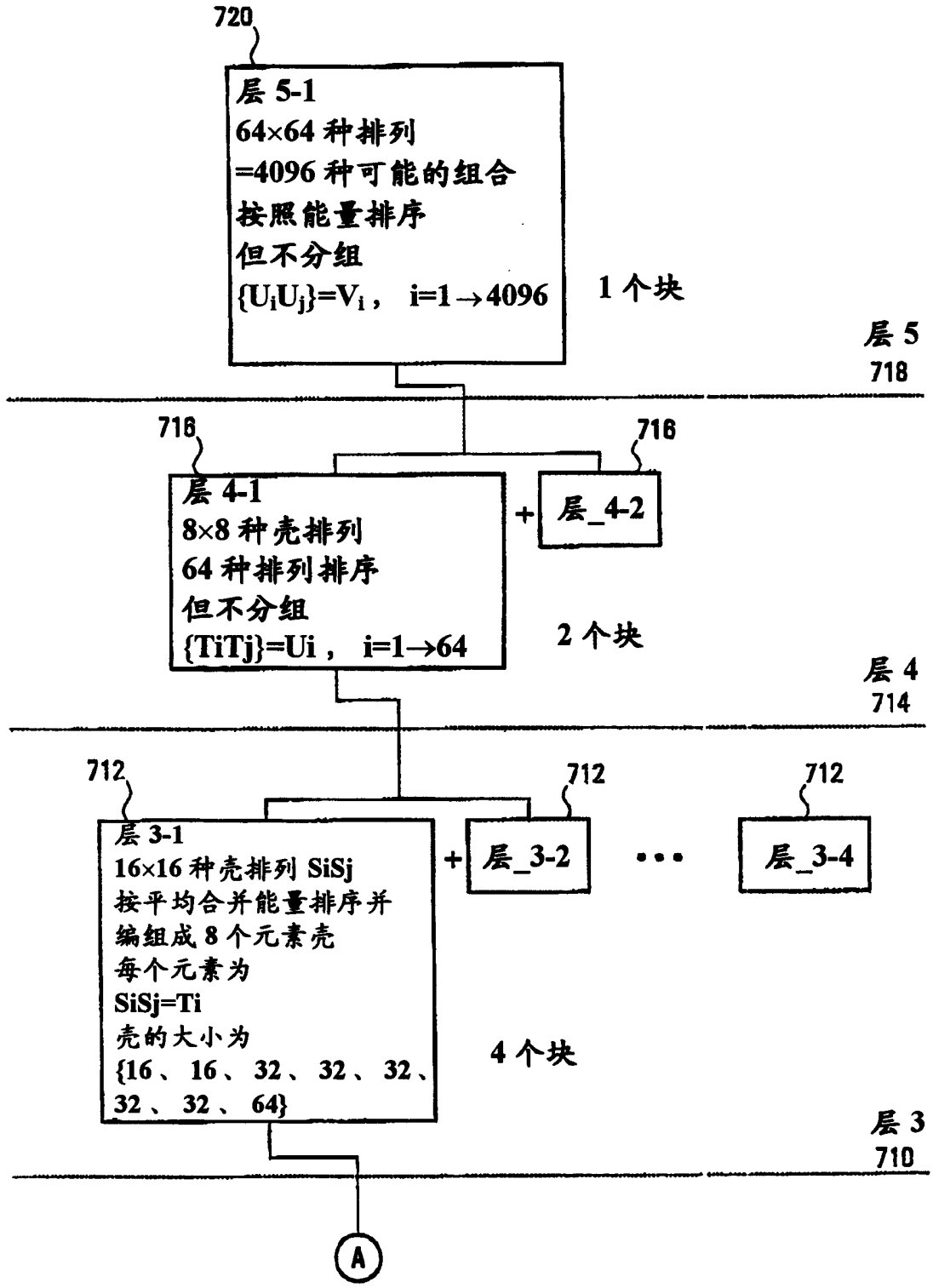

720
层 5-1
64×64 种排列
=4096 种可能的组合
按照能量排序
但不分组
{UiUj}=Vi， i=1→4096
1 个块
层 5
718
716
716
层 4-1
8×8 种壳排列
64 种排列排序
但不分组
{TiTj}=Ui， i=1→64
+
层_4-2
2 个块
层 4
714
712
712
712
层 3-1
16×16 种壳排列 SiSj
按平均合并能量排序并
编组成 8 个元素壳
每个元素为
SiSj=Ti
壳的大小为
{16、16、32、32、32、
32、32、64}
+
层_3-2
• • •
层_3-4
4 个块
层 3
710
A

图 7A

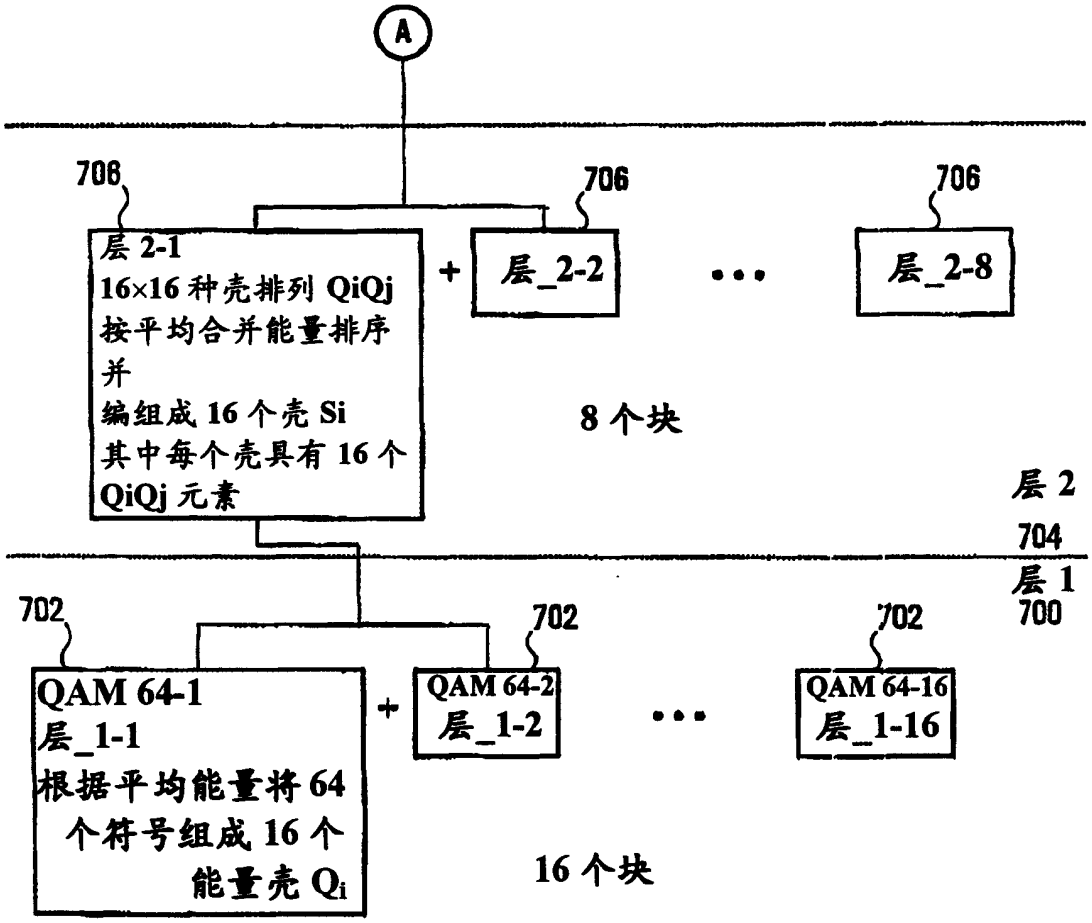

A
706
706
706
层 2-1
16×16 种壳排列 QiQj
按平均合并能量排序
并
编组成 16 个壳 Si
其中每个壳具有 16 个
QiQj 元素
+
层_2-2
• • •
层_2-8
8 个块
层 2
704
层 1
700
702
702
702
QAM 64-1
层_1-1
根据平均能量将 64
个符号组成 16 个
能量壳 Qi
+
QAM 64-2
层_1-2
• • •
QAM 64-16
层_1-16
16 个块

图 7A
(续)

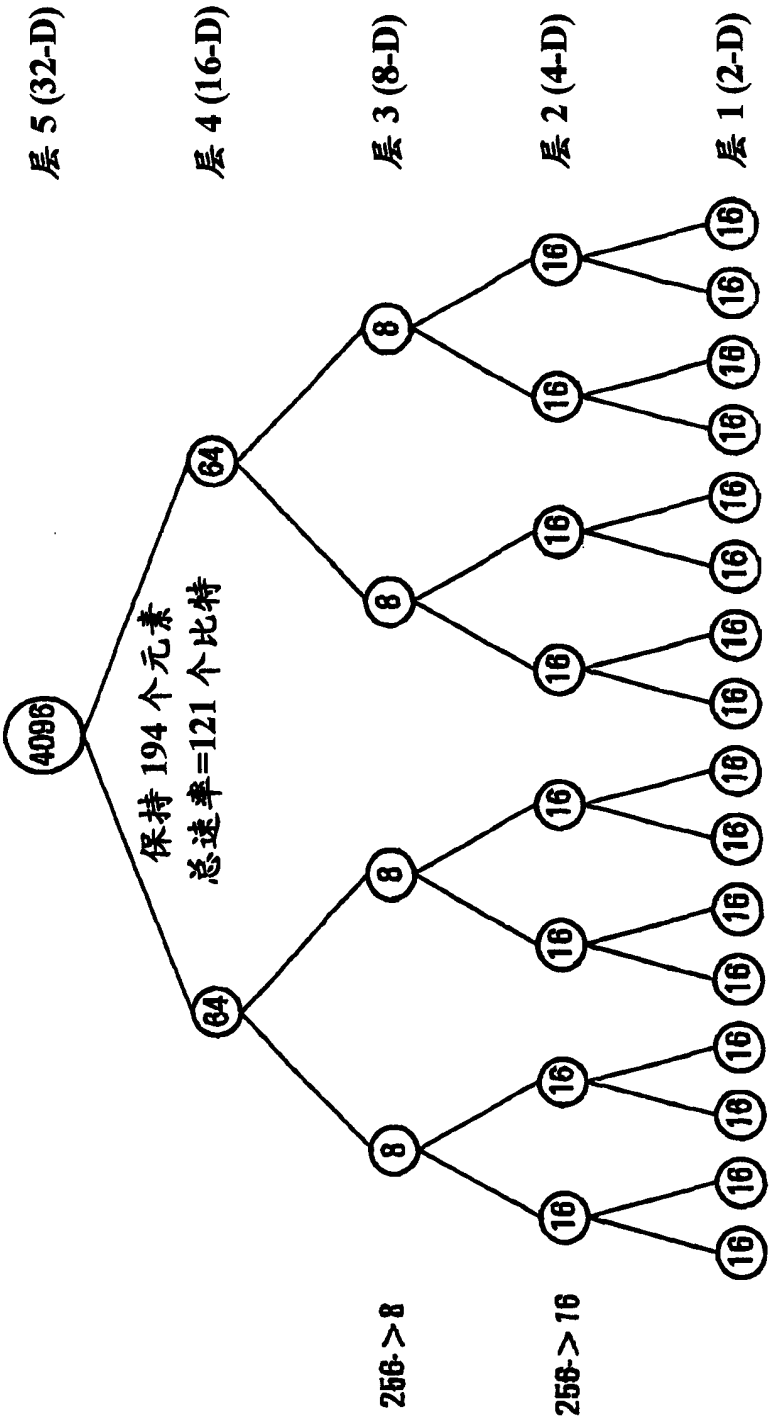
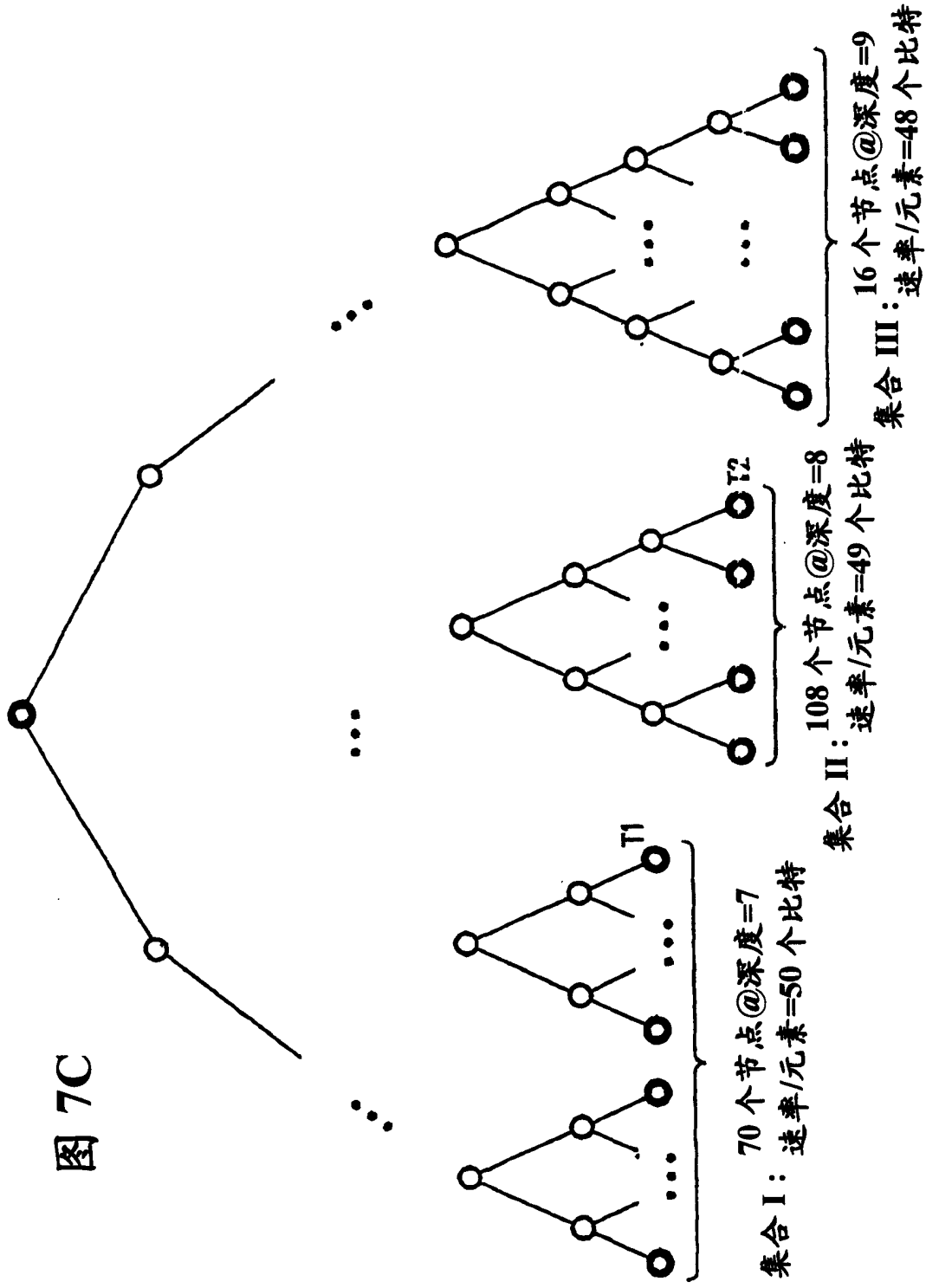

层 5 (32-D)
层 4 (16-D)
层 3 (8-D)
层 2 (4-D)
层 1 (2-D)
4096
保持 194 个元素
总速率=121 个比特
64
64
8
8
8
8
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
16
256->8
256->16

图 7B

图 7C


T1
T2
集合 I：70 个节点@深度=7
速率/元素=50 个比特
集合 II：108 个节点@深度=8
速率/元素=49 个比特
集合 III：16 个节点@深度=9
速率/元素=48 个比特

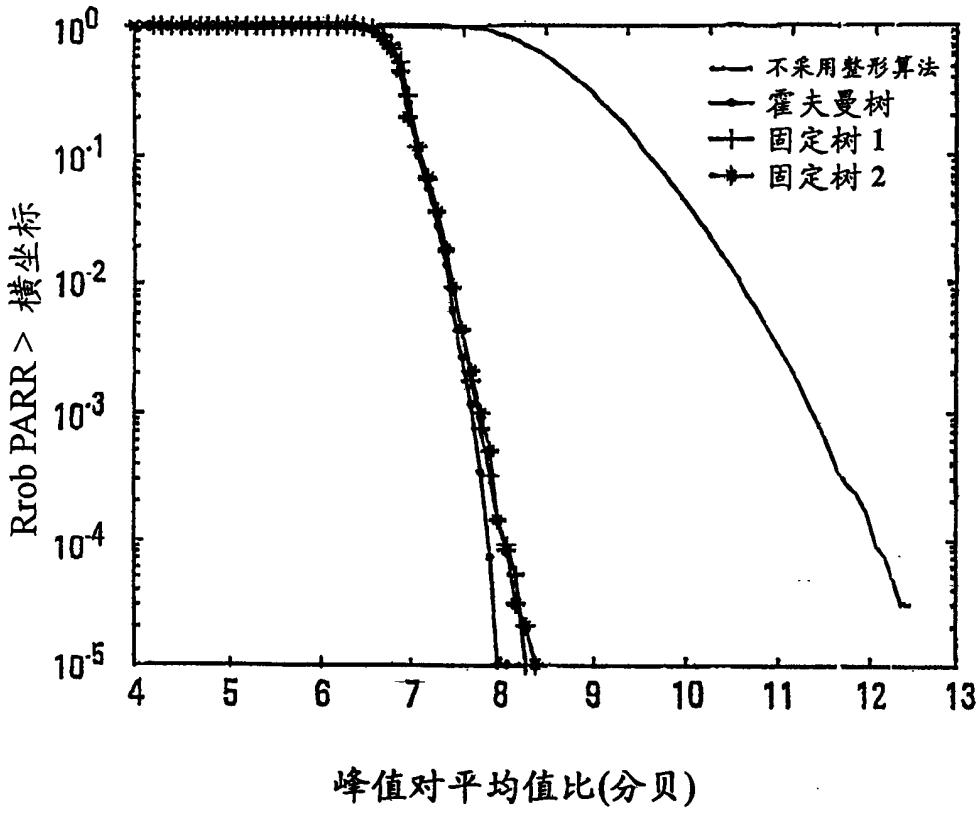

Rrob PARR > 横坐标
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
4
5
6
7
8
9
10
11
12
13
不采用整形算法
霍夫曼树
固定树 1
固定树 2
峰值对平均值比(分贝)

图 8

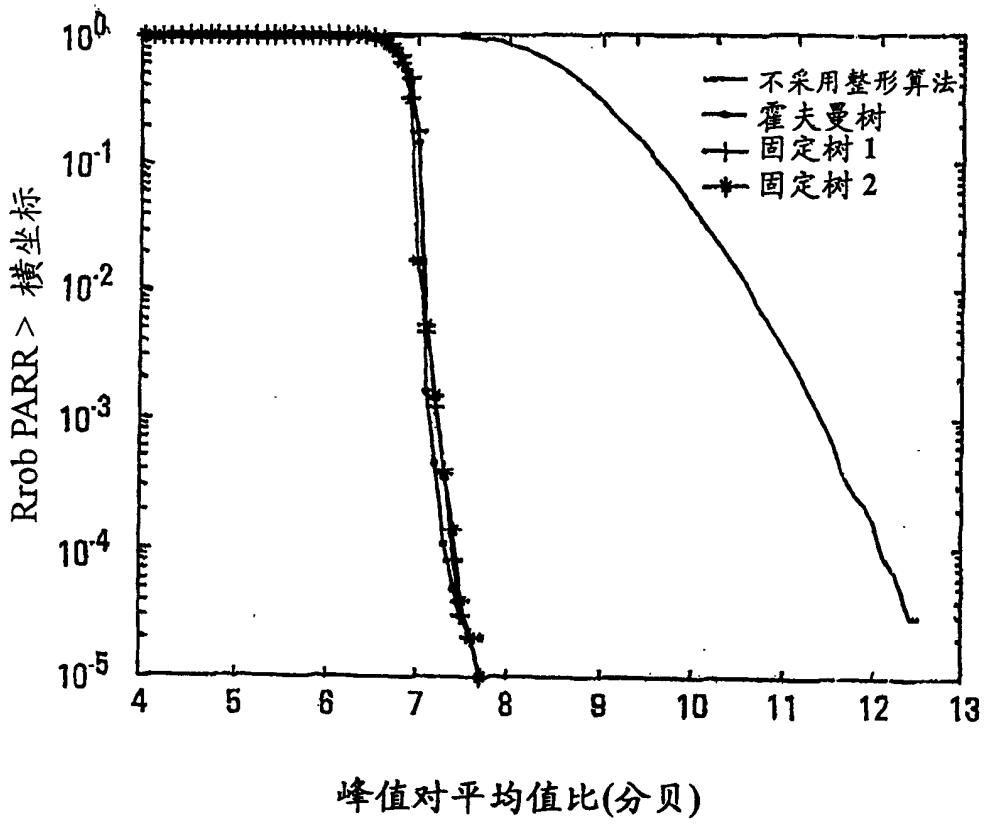

Rrob PARR > 横坐标
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
4
5
6
7
8
9
10
11
12
13
不采用整形算法
霍夫曼树
固定树 1
固定树 2
峰值对平均值比(分贝)

图 9

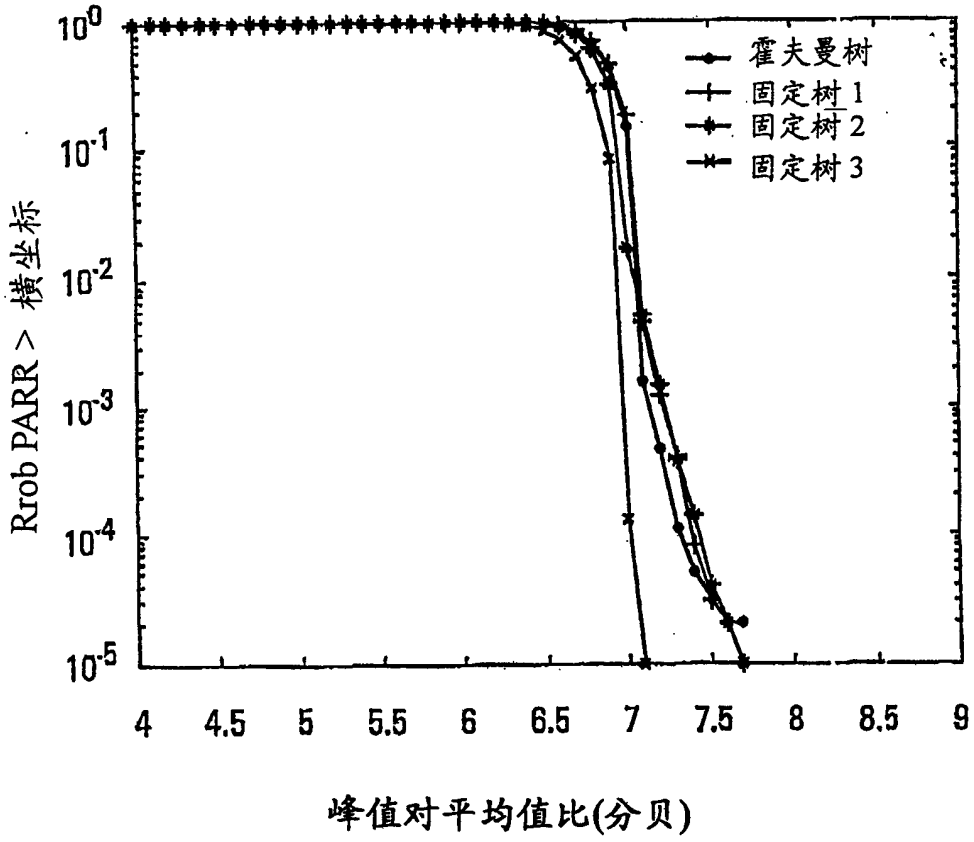

霍夫曼树
固定树 1
固定树 2
固定树 3
Rrob PARR > 横坐标
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
4
4.5
5
5.5
6
6.5
7
7.5
8
8.5
9
峰值对平均值比(分贝)

图 10