



(12) 发明专利

(10) 授权公告号 CN 101632236 B

(45) 授权公告日 2013.06.26

(21) 申请号 200880007948.2

代理人 刘瑜 王英

(22) 申请日 2008.03.20

(51) Int. Cl.

(30) 优先权数据

H04L 27/00(2006.01)

60/896,040 2007.03.21 US

12/042,834 2008.03.05 US

(56) 对比文件

US 7031373 B1, 2006.04.18, 第5-6段, 第12-14段, 第20、22段, 图2.

(85) PCT申请进入国家阶段日

US 6526090 B1, 2003.02.25, 全文.

2009.09.11

US 5490165 A, 1996.02.06, 全文.

(86) PCT申请的申请数据

PCT/US2008/057692 2008.03.20

JP 2001211079 A, 2001.08.03, 全文.

(87) PCT申请的公布数据

W02008/116086 EN 2008.09.25

审查员 易水英

(73) 专利权人 高通股份有限公司

地址 美国加利福尼亚

(72) 发明人 M·蔡 J·J·吴 P·斯里尼瓦桑

R·西尼尔 V·乔迪亚

(74) 专利代理机构 永新专利商标代理有限公司

72002

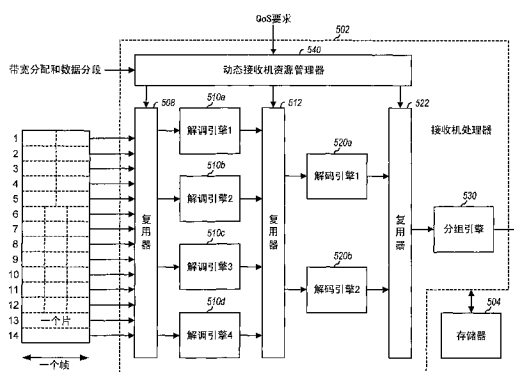
权利要求书3页 说明书12页 附图14页

(54) 发明名称

接收机资源的动态管理

(57) 摘要

描述了用于动态管理接收机资源以获得良好的延迟和吞吐量性能的技术。接收机可以具有各种类型的接收机资源,诸如用于解调分组的解调资源、用于解码分组的解码资源、分组处理资源等。这些各种类型的接收机资源可以被动态地分配给分组。在一个设计方案中,可以估计接收机要处理的每一个分组的接收机资源使用量。可以基于每一个分组的估计的接收机资源使用量和一个或多个诸如优先级、延迟要求等的分配标准将可用的接收机资源分配给分组。根据接收机资源分配可以确定每一个分组的延迟。如果有任何分组的延迟要求没有被满足,可以跳过或推迟一个或多个分组的接收机处理,从而使得每一个要处理的分组的延迟要求都能被满足。



1. 一种用于管理接收机资源的装置,包括:
至少一个处理器,其特征在进一步包括:
管理器,配置成基于分组的调制阶数、带宽量、时间量、片的数目以及层的数目中的至少一个估计至少一个分组中每一个分组的接收机资源使用量,每一个分组的所述接收机资源使用量是用于处理所述分组的接收机资源的数量,并且基于每一个分组的所述估计的接收机资源使用量将可用的接收机资源分配给所述至少一个分组,以及
存储器,其耦合到所述至少一个处理器。
2. 根据权利要求 1 所述的装置,其中,所述可用的接收机资源包括解调资源,并且其中,所述管理器用于估计每一个分组的解调资源使用量,并且基于每一个分组的所述估计的解调资源使用量将可用的解调资源分配给所述至少一个分组,其中,所述解调资源使用量是用于解调所述分组的解调资源的数量。
3. 根据权利要求 2 所述的装置,其中,所述管理器用于基于带宽量、时间量、层的数目以及所述分组的调制阶数中的至少一个来估计每一个分组的所述解调资源使用量。
4. 根据权利要求 2 所述的装置,其中,每一个分组在至少一个片中被发送,并且其中,所述管理器用于基于用于所述分组的片的数目来估计每一个分组的所述解调资源使用量。
5. 根据权利要求 2 所述的装置,其中,所述可用的解调资源包括可并行工作的多个解调引擎。
6. 根据权利要求 5 所述的装置,其中,所述至少一个分组在多个片中被传送,并且其中,所述管理器用于分配近似等于片的数目的多个解调引擎。
7. 根据权利要求 5 所述的装置,其中,所述管理器用于基于分组的大小从最小到最大将所述多个解调引擎分配给所述至少一个分组。
8. 根据权利要求 5 所述的装置,其中,所述管理器用于基于分组的大小从最大到最小将所述多个解调引擎分配给所述至少一个分组。
9. 根据权利要求 1 所述的装置,其中,所述可用的接收机资源包括解码资源,并且其中,所述管理器用于估计每一个分组的解码资源使用量,并所述管理器基于每一个分组的所述估计的解码资源使用量将可用的解码资源分配给所述至少一个分组,其中,所述解码资源使用量是用于解码所述分组的解码资源的数量。
10. 根据权利要求 9 所述的装置,其中,所述管理器用于基于分组的大小和所述分组的解码迭代次数中的至少一个来估计每一个分组的所述解码资源使用量。
11. 根据权利要求 9 所述的装置,其中,每一个分组包括至少一个子分组,并且其中,所述管理器用于基于所述分组的子分组数目来估计每一个分组的解码资源使用量。
12. 根据权利要求 9 所述的装置,其中,所述管理器用于基于每一个分组的正确解码概率,从具有最高正确解码概率的分组开始按顺序将所述可用的解码资源分配给所述至少一个分组。
13. 根据权利要求 9 所述的装置,其中,所述至少一个分组包括多个子分组,并且其中,所述管理器用于基于针对每一个子分组所接收到的传输次数来将所述可用的解码资源分配给所述多个子分组。
14. 根据权利要求 9 所述的装置,其中,所述可用的解码资源包括多个可并行工作的解码引擎。

15. 根据权利要求 1 所述的装置,其中,所述可用的接收机资源包括分组处理资源,并且其中,所述管理器用于估计每一个分组的分组处理资源使用量,并所述管理器基于每一个分组的所述估计的分组处理资源使用量将可用的分组处理资源分配给所述至少一个分组,其中,所述分组处理资源使用量是用于处理所述分组的分组处理资源的数量。

16. 根据权利要求 1 所述的装置,其中,所述管理器用于基于每一个分组的延迟要求,从具有最短延迟要求或最大延迟敏感性的分组开始按顺序将所述可用的接收机资源分配给所述至少一个分组。

17. 根据权利要求 1 所述的装置,其中,所述管理器用于基于每一个分组的优先级,从具有最高优先级的分组开始按顺序将所述可用的接收机资源分配给所述至少一个分组。

18. 根据权利要求 1 所述的装置,其中,所述管理器用于跳过或推迟在所述至少一个分组中的一个或多个分组的处理,以使得要处理的每一个分组都满足该分组的要求,所述管理器还用于基于每一个分组的延迟要求、优先级、正确解码的可能性、传输次数、片的数目以及大小中的至少一个来选择所述一个或多个分组以跳过或推迟处理。

19. 一种管理接收机资源的方法,包括:

基于分组的调制阶数、带宽量、时间量、片的数目以及层的数目中的至少一个,估计至少一个分组中每一个分组的接收机资源使用量,每一个分组的所述接收机资源使用量是用于处理所述分组的接收机资源的数量;以及

基于每一个分组的所述估计的接收机资源使用量将可用的接收机资源分配给所述至少一个分组。

20. 根据权利要求 19 所述的方法,其中,所述估计接收机资源使用量包括估计每一个分组的解调资源使用量,所述解调资源使用量是用于解调所述分组的解调资源的数量,并且其中,所述分配可用的接收机资源包括基于每一个分组的所述估计的解调资源使用量来将可用的解调资源分配给所述至少一个分组。

21. 根据权利要求 20 所述的方法,其中,所述估计每一个分组的解调资源使用量包括基于带宽量、时间的量、片的数目、层的数目以及所述分组的调制阶数中的至少一个来估计每一个分组的所述解调资源使用量。

22. 根据权利要求 19 所述的方法,其中,所述估计接收机资源使用量包括估计每一个分组的解码资源使用量,所述解码资源使用量是用于解码所述分组的解码资源的数量,并且其中,所述分配可用的接收机资源包括基于每一个分组的所述估计的解码资源使用量来将可用的解码资源分配给所述至少一个分组。

23. 根据权利要求 22 所述的方法,其中,所述估计每一个分组的解码资源使用量包括基于分组大小、解码迭代的次数、所述分组的子分组数目中的至少一个来估计每一个分组的所述解码资源使用量。

24. 根据权利要求 19 所述的方法,其中,所述分配可用的接收机资源包括基于每一个分组的延迟要求或优先级,从具有最短延迟要求或最高优先级的分组开始按顺序将所述可用的接收机资源分配给所述至少一个分组。

25. 根据权利要求 19 所述的方法,还包括:

跳过或推迟在所述至少一个分组中的一个或多个分组的处理,以使得要处理的每一个分组都满足该分组的要求;以及

基于每一个分组的延迟要求、优先级、正确解码的可能性、传输的次数、片的数目以及大小中的至少一个来选择所述一个或多个分组以跳过或推迟处理。

26. 一种用于管理接收机资源的装置,包括:

估计模块,用于基于分组的调制阶数、带宽量、时间量、片的数目以及层的数目中的至少一个估计至少一个分组中每一个分组的接收机资源使用量,每一个分组的所述接收机资源使用量是用于处理所述分组的接收机资源的数量;以及

分配模块,用于基于每一个分组的所述估计的接收机资源使用量将可用的接收机资源分配给所述至少一个分组。

27. 根据权利要求 26 所述的装置,其中,

所述估计模块包括用于估计每一个分组的解调资源使用量的模块,所述解调资源使用量是用于解调所述分组的解调资源的数量,并且其中,所述分配模块包括用于基于每一个分组的所述估计的解调资源使用量来将可用的解调资源分配给所述至少一个分组的模块。

28. 根据权利要求 27 所述的装置,其中,所述用于估计每一个分组的解调资源使用量的模块包括用于基于带宽量、时间的量、片的数目、层的数目、所述分组的调制阶数中的至少一个来估计每一个分组的所述解调资源使用量的模块。

29. 根据权利要求 26 所述的装置,其中,所述估计模块包括用于估计每一个分组的解码资源使用量的模块,所述解码资源使用量是用于解码所述分组的解码资源的数量,并且其中,所述分配模块包括用于基于每一个分组的所述估计的解码资源使用量来将可用的解码资源分配给所述至少一个分组的模块。

30. 根据权利要求 29 所述的装置,其中,所述用于估计每一个分组的解码资源使用量的模块包括用于基于分组大小、解码迭代的次数、所述分组的子分组数目中的至少一个来估计每一个分组的所述解码资源使用量的模块。

31. 根据权利要求 26 所述的装置,其中,所述分配模块包括用于基于每一个分组的延迟要求或优先级,从具有最短延迟要求或最高优先级的分组开始按顺序将所述可用的接收机资源分配给所述至少一个分组的模块。

32. 根据权利要求 26 所述的装置,还包括:

跳过或推迟模块,用于跳过或推迟在所述至少一个分组中的一个或多个分组的处理,以使得要处理的每一个分组都满足该分组的要求;以及

选择模块,用于基于每一个分组的延迟要求、优先级、正确解码的可能性、传输的次数、片的数目以及大小中的至少一个来选择所述一个或多个分组以跳过或推迟处理。

接收机资源的动态管理

[0001] 本申请要求于 2007 年 3 月 21 日递交的、名称为“DYNAMIC RECEIVERRESOURCE MANAGEMENT FOR BROADBAND DATA MODEMS”的美国临时申请 No. 60/896, 040 的优先权, 该临时申请已经转让给本申请的受让人, 故以引用方式将其全部内容明确地并入本文。

技术领域

[0002] 本公开整体上涉及通信, 并且更具体地涉及用于管理接收机资源的技术。

背景技术

[0003] 无线通信系统被广泛地部署以提供诸如语音、视频、分组数据、消息传递、广播等各种通信业务。这些无线系统可以是能够通过共享可用的系统资源以支持多个用户的多址系统。这种多址系统的实例包括码分多址 (CDMA) 系统、时分多址 (TDMA) 系统、频分多址 (FDMA) 系统、正交 FDMA (OFDMA) 系统、单载波 FDMA (SC-FDMA) 系统。

[0004] 在无线通信系统中, 基站可以向终端发送大量的数据。在终端的宽带接收机可以对从基站接收到的传输进行解调和解码以恢复发送到终端的数据。所述接收机用于对接收到的传输进行解调和解码的资源可能是有限的。所述接收机用于处理接收到的传输以及恢复数据的时间量被称为接收机的延迟。接收机的延迟和吞吐量可能取决于多种因素, 例如可用的接收机资源数量、发送到终端的数据量、数据是如何被发送到终端的, 等等。通常, 希望使用可用的接收机资源以尽可能高效率地和尽可能快速地处理接收到的传输, 从而满足延迟和吞吐量需求。

发明内容

[0005] 本文描述了用于动态管理接收机资源以获得良好的延迟和吞吐量性能的技术。接收机可以具有多种类型的接收机资源, 例如用于对分组进行解调的解调 (demod) 资源、用于对分组进行解码的解码资源、分组处理资源等等。这些多种类型的接收机资源可以被动态地分配给分组, 例如, 以逐帧为基础的方式。

[0006] 在动态接收机资源管理的一个设计方案中, 可以估计接收机要处理的至少一个分组中的每一个分组的接收机资源使用量。可以基于用于分组的带宽容量和 / 或时间量、用于分组的片 (tile) 的数目、分组传送所经过的层的数目、分组的调制阶数等来估计每一个分组的解调资源使用量。可以基于分组长度、为分组执行的解码迭代的次数、用于分组的编码方案等来估计每一个分组的解码资源使用量。

[0007] 可以基于所估计的每一个分组的接收机资源使用量和一个或多个分配标准来将可用的接收机资源分配给至少一个分组。在一个设计方案中, 可以按照顺序将可用的接收机资源分配给至少一个分组, 其中可以基于每一个分组的优先级、每一个分组的延迟要求等来确定上述顺序。可以基于每一个分组所使用的片的数目将可用的解调资源分配给至少一个分组。可以基于分组的大小、每一个分组中的子分组的数目、正确解码每一个子分组的可能性等将可用的解码资源分配给至少一个分组。可以基于初始的接收机资源分配确定每

一个分组的延迟。如果有任何分组的延迟要求不满足,那么可以跳过或推迟对一个或多个分组/子分组的接收机处理(例如,解调和/或解码),以使得每一个需要处理的分组的延迟要求能够被满足。然后可以按照初始的或经修改的接收机资源分配的指示来调度接收机处理至少一个分组。

[0008] 会在下文中对本公开的各个方面和特点作更详细的描述。

附图说明

- [0009] 图 1 示出了无线通信系统。
- [0010] 图 2 示出了在使用 HARQ 的前向链路上的数据传输。
- [0011] 图 3 示出了帧结构。
- [0012] 图 4 示出了基站和终端的框图。
- [0013] 图 5 示出了接收机处理器的框图。
- [0014] 图 6A 示出了在 14 个片中的 3 个分组的传输。
- [0015] 图 6B 示出了将片近似平均分配给解调引擎。
- [0016] 图 6C 示出了基于分组的大小从最小到最大将片分配给解调引擎。
- [0017] 图 6D 示出了基于分组的大小从最大到最小将片分配给解调引擎。
- [0018] 图 7 示出了针对图 6B 的分配所进行的解码和分组处理。
- [0019] 图 8 示出了针对图 6C 的分配所进行的解码和分组处理。
- [0020] 图 9 和 10 示出了针对图 6D 的分配所进行的解码和分组处理的两个实例。
- [0021] 图 11 示出了用于动态管理接收机资源的过程。
- [0022] 图 12 示出了用于动态管理接收机资源的装置。
- [0023] 图 13 示出了具有动态接收机资源管理的接收机处理器的框图。

具体实施方式

[0024] 本文描述的技术可以被用于诸如 CDMA、TDMA、FDMA、OFDMA 和 SC-FDMA 系统的各种无线通信系统。术语“系统”和“网络”通常可互换使用。CDMA 系统可以实现诸如 cdma2000、通用陆地无线接入 (UTRA) 等的无线技术。OFDMA 系统可以实现诸如超移动宽带 (UMB)、演进 UTRA (E-UTRA)、IEEE 802. 16、IEEE 802. 20、Flash-OFDM® 等的无线技术。在名为“第三代合作伙伴计划” (3GPP) 的组织文件中描述了 UTRA 和 E-UTRA。在名为“第三代合作伙伴计划 2” (3GPP2) 的组织文件中描述了 cdma2000 和 UMB。这些各种无线技术和标准是本领域公知的。为了清楚起见,下面针对 UMB 描述该技术的特定方面,在下面说明书中的许多地方使用的是 UMB 术语。在公众可获得的 2007 年 8 月发布的名称为“超移动宽带 (UMB) 物理层空中接口规范”的 3GPP2C. S0084-001 中描述了 UMB。

[0025] 图 1 示出了无线通信系统 100,其也可以被称为接入网络 (AN)。为了简明,在图 1 中仅示出了一个基站 110 和两个终端 120 和 130。基站是与终端通信的站。基站也可以被称为接入点、节点 B、演进节点 B 等。终端可以是固定的或移动的,并且也可以被称为接入终端 (AT)、移动台、用户设备、用户单元、站等。终端可以是蜂窝电话、个人数字助理 (PDA)、无线通信设备、无线调制解调器、手持设备、膝上型计算机、无绳电话等。终端可以在前向和/或反向链路上与基站进行通信。前向链路(或下行链路)是指从基站到终端的通信链路,

反向链路（或上行链路）是指从终端到基站的通信链路。在图 1 中，终端 120 可以经由前向链路 122 从基站 100 接收数据并可以经由反向链路 124 发送数据。终端 130 可以经由向链路 132 从基站 100 接收数据并可以经由反向链路 134 发送数据。本文描述的技术可以被用于在前向链路以及反向链路上接收数据。

[0026] 系统可以支持以分组的方式传送数据。分组可以具有可变大小，该大小可以取决于多种因素，诸如要发送的数据量、可用于发送分组的无线资源的量、信道条件等。在一个设计方案中，分组可以被划分为 N_{SP} 个子分组，其中 $N_{SP} \geq 1$ 并可以以如下方式确定：

$$[0027] \quad N_{SP} = \left\lceil \frac{\text{分组大小}}{\text{最大的子分组大小}} \right\rceil, \text{公式 (1)}$$

[0028] 其中“ $\lceil \cdot \rceil$ ”代表上取整运算符。所述最大的子分组大小可以由系统定义的固定值。每一个子分组可以被发射机单独编码并被接收机单独解码。在另一个设计方案中，不使用子分组，每一个分组可以被单独编码和解码。本文描述的技术可应用于上述两种设计方案中，并且无论分组或子分组是否被单独编码和解码均可以应用。然而，为了清楚起见，下面的主要描述是针对分组被划分为 N_{SP} 个子分组的设计方案。

[0029] 系统可以支持混合自动重传 (HARQ)。对于在前向链路上的 HARQ，基站可以为一个子分组发送一个或多个传输，直到该子分组被终端正确解码，或者达到已经发送传输的最大数目、或者遇到其他终止条件。HARQ 可以提高数据传输的可靠性。

[0030] 图 2 示出了在使用 HARQ 的前向链路上的数据传输。基站 110 可以在帧 n 中处理子分组 1 并为子分组 1 发送第一个 HARQ 传输（传输 1）。HARQ 传输是在一个帧中针对一个子分组的传输。终端 120 可能接收第一个 HARQ 传输，对子分组 1 进行解码时发生错误并发送否定应答 (NAK)。基站 110 可以接收该 NAK 并在帧 $n+Q$ 中为子分组 1 发送第二个 HARQ 传输（传输 2），其中 Q 可以是固定的或可配置的值。终端 120 可以接收该第二个 HARQ 传输，根据第一个和第二个 HARQ 传输正确解码子分组 1，并发送确认应答 (ACK)。基站 110 可以接收该 ACK 并按照相同的方式处理和发送子分组 2。

[0031] 可以定义 Q 个 HARQ 交织，其中每一个 HARQ 交织包括被 Q 个帧间隔开的帧。基站 110 可以在多达 Q 个 HARQ 交织上将多个子分组并行地发送到终端 120，其中每一个子分组在一个 HARQ 交织上发送。可以针对每一个子分组在用于该分组的 HARQ 交织上发送一个或多个 HARQ 传输。每一个子分组可以被处理（例如，被编码和被调制）以使得使用目标次数的 HARQ 传输能够正确地解码该子分组，目标次数可以是 2、3、4 等。

[0032] 系统可以使用正交频分复用 (OFDM) 和 / 或单载波频分复用 (SC-FDM)。OFDM 和 SC-FDM 将系统的带宽分割成多个 (N_{FFT}) 正交子载波，其通常也被称为音调 (tone)、仓 (bin) 等。每一个子载波可以用数据进行调制。通常，调制符号在频域使用 OFDM 进行发送，在时域使用 SC-FDM 进行发送。相邻子载波之间的距离可以是固定的，并且子载波的数量可以取决于系统带宽。

[0033] 图 3 示出了可以用于前向链路的超帧结构 300 的设计方案。传输的时间轴可以被划分为多个超帧单元。每一个超帧可以跨越特定的时间长度，该时间长度可以是固定的或可配置的。每一个超帧可以包括跟随有 F 个帧的前导 (preamble)，其中 F 可以是任意整数值。每一个帧可以覆盖 N_{FRAME} 个 OFDM 符号周期，其中 N_{FRAME} 可以等于 4、6、或 8 或某个其他数值。超帧的前导可以携带系统信息和获取导频。每一个帧可以携带业务数据、控制信息 /

信令、导频等。

[0034] 前向链路可用的时间频率资源可以被划分为片 (tile)。片是时间频率资源的块, 并且可以被称为资源块、时间频率块等。片可以覆盖任意的时间和频率维度, 并且可以是固定的或可配置的。在一个设计方案中, 片覆盖一个帧中的 16 个子载波。每一个帧可以包括 L 个片, 其中 L 可以取决于子载波的总数 (N_{FFT})。

[0035] 图 4 示出了在图 1 中的基站 110 和终端 120 的设计框图。在这个设计方案中, 基站 110 装配有 S 个天线 424a-424s, 终端 120 装配有 T 个天线 452a-452t, 其中通常 $S \geq 1$ 并且 $T \geq 1$ 。基站 110 和终端 120 可以支持单输入单输出 (SISO)、单输入多输出 (SIMO)、多输入单输出 (MISO)、和 / 或多输入多输出 (MIMO) 传输。

[0036] 在前向链路上, 在基站 110 中, 发射机处理器 410 可以从数据源 408 接收要发送给终端 120 的数据分组, 并可以将分组分割成子分组。发射机处理器 410 然后可以处理 (例如, 编码和调制) 每一个子分组以获得相应的输出子分组并将输出子分组映射到分配给终端 120 的片。发射机处理器 410 还可以将导频符号映射到被分配的片, 在可应用的条件下执行 MIMO 处理, 并且向 S 个发射机单元 (TMTR) 422a-422s 提供 S 个输出符号流。每一个发射机单元 422 可以处理自身的输出符号流 (例如, 为 OFDM) 以获得输出码片流。每一个发射机单元 422 可以进一步调节 (例如, 转换为模拟信号、滤波、放大、上变频) 它的输出码片流以产生前向链路信号。可以通过 S 个天线 424a-424s 分别发射来自发射机单元 422a-422s 的 S 个前向链路信号。

[0037] 在终端 120, T 个天线 452a-452t 可以从基站 110 接收前向链路信号, 每一个天线 452 可以向关联的接收机单元 (RCVR) 454 提供接收的信号。每一个接收机单元 454 可以调节 (例如, 滤波、放大、下变频和数字化) 其所接收的信号以获得采样, 对采样进行处理 (例如, 为 OFDM) 以获得接收的符号, 并将接收的符号提供给接收机处理器 460。接收机处理器 460 可以对在分配给终端 120 的片中的所接收的符号执行解调, 从分配的片解映射子分组, 解码每一个子分组, 执行分组处理, 并将被解码的分组提供给数据接收器 462。通常, 接收机处理器 460 的处理与基站 110 中的发射机处理器 410 所进行的处理是互逆的。

[0038] 在反向链路上, 在终端 120, 发射机处理器 480 可以从数据源 478 接收分组, 将该分组分割成子分组, 对每一个子分组进行编码以获得输出子分组, 并将输出子分组映射到分配给终端 120 的片, 以进行上行链路传输。发射机处理器 480 还可以将导频符号映射到被分配的片中, 在可应用的条件下执行空间处理, 并提供 T 个输出符号流。发射机单元 454a-454t 可以处理 T 个输出符号流以获得 T 个可以通过天线 452a-452t 发送的反向链路信号。在基站 110, 来自终端 120 的反向链路信号可以被天线 424a-424s 接收, 被接收机单元 422a-422s 处理, 并进一步被接收机处理器 440 处理以恢复出终端 120 所传送的分组。

[0039] 控制器 / 处理器 430 和 470 可以分别指示在基站 110 和终端 120 处的操作。存储器 432 和 472 可以分别为基站 110 和终端 120 存储数据和程序代码。调度器 434 可以调度终端 120 以用于在前向和 / 或反向链接上的数据传输, 并可以为数据传输分配诸如片之类的无线电资源。

[0040] 接收机处理器也可以被称为是接收机、宽带接收机、宽带数据调制解调器等, 并且这些各种术语可以互换使用。通常, 接收机处理器可以包括用于诸如解调、解码、分组处理等之类的任何功能组的资源。接收机处理器还可以包括用于每一个功能的任意数目的资

源,资源数目可以是固定的或可配置的。在一个设计方案中,接收机处理器可以包括用于每一个功能的固定的资源数目,例如,用于每一个功能的固定数目的处理引擎。在另一个设计方案中,接收机处理器可以包括可用于执行不同功能的固定数目的资源(例如,固定数目的处理引擎)。在这个设计方案中,接收机处理器可以被配置为根据要执行的处理为不同的功能包括不同数目的资源。为了清楚,下面的多数描述针对接收机处理器对于每个功能包括固定数目的资源的设计方案。

[0041] 图5示出了接收机处理器500的设计框图,该接收机处理器可用于图4中的接收机处理器440和460中的每一个。在这个设计方案中,接收机处理器500包括4个解调(demod)引擎510a-510d,2个解码引擎520a-520b,1个分组引擎530。在下面的描述中,解调引擎510a-510d也被称为解调引擎1-4,解码引擎520a-520b也被称为解码引擎1和2。通常,接收机处理器为每一个功能可以包括任意数目的引擎。接收机处理器还可以包括用于图5中未示出的其他功能的引擎。

[0042] 在接收机处理器500中,复用器(Mux)508可以从所有分配的片中获得接收的符号,并且可以将来自每一个片的接收的符号提供给为该片分配的解调引擎510。在一个设计方案中,每一个解调引擎510可以在每一个时间单元开始处理一个新的片,并可以用下面描述的流水线方式处理多个片。解调引擎510a-510d可以并行处理多达4个片。每一个解调引擎510可以执行诸如解扰、信道估计、MIMO检测、相关检测、对数似然比(LLR)计算等的任务。

[0043] 复用器512可以从所有的解调引擎510a-510d中获得被解调的符号,并可以将每一个子分组的被解调的符号提供给为该子分组分配的解码引擎520。在一个设计方案中,每一个解码引擎520可以一次解码一个子分组并当待处理的子分组已经被解码时可以开始处理新的子分组。解码引擎520a-520b可以并行解码直到2个子分组。每一个解码引擎520可以执行诸如合并、解映射、解交织、解码、CRC校验等的任务。

[0044] 复用器522可以从两个解码引擎520a和520b获得被解码的子分组并将每一个分组的被解码的子分组提供给分组引擎530。在一个设计方案中,分组引擎530可以一次处理一个分组,并且只要接收到分组的所有被解码的子分组就可以开始处理该分组。分组引擎530可以执行诸如重新组装、复制检测、路由等的任务。

[0045] 图5示出了接收机处理器500的特定设计方案。接收机处理器500还可以用其他设计方案实现,并可以为每一个功能包括不同数目的引擎。存储器504可以为接收机处理器500存储数据和程序代码。

[0046] 分组的延迟是接收机处理器处理分组所需要的时间量。不同的接收机处理器由于它们配置和能力的不同,对于同样传输的数据可能具有不同的延迟。另一方面,给定的接收机处理器由于带宽分配和数据分段的不同,对于相同数量的不同数据可能具有不同的延迟。这将用下面的实例进行说明。

[0047] 图6A-6D示出了分配4个解调引擎以对在14个片内的3个分组的传输进行解调的3个方案。在下面的实例中,假设对于每个片的解调时间是相同的。解调引擎能够在每一个时间单元中开始处理新的片,其中时间单元是任意适合的持续时间。一旦用于分组的所有片已经被处理了至少一个时间单元,那么该分组的所有子分组就准备好进行解码。

[0048] 图6A示出了在14个片内的3个分组的传输的实例。在这个实例中,分组1被分

为 2 个子分组（被标记为 SP1 和 SP2）并在 5 个片 1-5 中进行发送。分组 2 被分为 3 个子分组（被标记为 SP3、SP4 和 SP5）并在 7 个片 6-12 中进行发送。分组 3 被分为 1 个分组（被标记为 SP6）并在 2 个片 13 和 14 中进行发送。为了清楚起见，图 6A 示出了每一个子分组占用了该子分组所属的分组所使用的每一个片的不同部分。为了获得分集，每一个分组中的所有子分组可以分布在该分组所使用的每一个片中。

[0049] 图 6B 示出了第一个解调资源分配方案，其中分组所使用的片尽可能平均地被分配给解调引擎。在图 6B 所给出的实施例中，最初的 4 个片 1-4 被分配给解调引擎 1，随后的 4 个片 5-8 被分配给解调引擎 2，随后的 3 个片 9-11 被分配给解调引擎 3，最后 3 个片 12-14 被分配给解调引擎 4。

[0050] 解调处理时间轴被分为时间单元并被标记为时刻 0、1、2、3、4 等。其中时刻 n 是从时刻 0 开始的 n 个时间单元。解调引擎 1 在时刻 0 开始处理片 1，随后在时刻 1 开始处理片 2，随后在时刻 2 开始处理片 3，随后在时刻 3 开始处理片 4。相类似地，解调引擎 2 分别在时刻 0、1、2 和 3 开始处理片 5、6、7 和 8。解调引擎 3 分别在时刻 0、1 和 2 开始处理片 9、10 和 11。解调引擎 4 分别在时刻 0、1 和 2 开始处理片 12、13 和 14。

[0051] 在图 6B 给出的实施例中，具有一个子分组的分组 3 被解调引擎 4 所处理并在 3 个时间单元之后准备解码。具有 2 个子分组的分组 1 被解调引擎 1 和 2 所处理并在 4 个时间单元之后准备解码。具有 3 个子分组的分组 2 被解调引擎 2、3 和 4 所处理并在 4 个时间单元之后准备解码。从而，分组 3、1 和 2 的解调延迟分别为 3 个、4 个和 4 个时间单元。

[0052] 图 6C 示出了第二个解调资源分配方案，其中基于分组的大小从最小到最大将分组所使用的片分配给解调引擎。在时刻 0，解调引擎 3 和 4 被分配用于最小的分组 3 的片 13 和 14，并且解调引擎 1 和 2 被分配用于第二小的分组 1 的片 1 和 2。在时刻 1，解调引擎 1、2 和 3 被分配用于分组 1 的片 3、4 和 5，并且解调引擎 4 被分配用于最大的分组 2 的片 6。在时刻 2，解调引擎 1、2、3 和 4 被分配用于分组 2 的片 7、8、9 和 10。在时刻 3，解调引擎 1 和 2 被分配用于分组 2 的片 11 和 12。

[0053] 对于图 6C 示出的实例，分组 3 被解调引擎 3 和 4 处理并在 1 个时间单元之后准备解码。分组 1 被解调引擎 1、2、3 和 4 处理并在 2 个时间单元之后准备解码。分组 2 被解调引擎 1、2、3 和 4 处理并在 4 个时间单元之后准备解码。分组 3、1 和 2 的解调延迟分别为 1 个、2 个和 4 个时间单元。

[0054] 图 6D 示出了第三个解调资源分配方案，其中基于分组的大小从最大到最小将用于分组的片分配给解调引擎，这与图 6C 中描述的第二个方案相反。在时刻 0，解调引擎 1、2、3 和 4 被分配用于最大的分组 2 的片 6、7、8 和 9。在时刻 1，解调引擎 1 被分配用于第二大的分组 1 的片 5，并且解调引擎 2、3 和 4 被分配用于分组 2 的片 10、11 和 12。在时刻 2，解调引擎 1、2、3 和 4 被分配用于分组 1 的片 1、2、3 和 4。在时刻 3，解调引擎 1 和 2 被分配用于最小的分组 3 的片 13 和 14。

[0055] 对于图 6D 中给出的实施例，分组 2 被解调引擎 1-4 处理并在 2 个时间单元之后准备解码。分组 1 被解调引擎 1-4 处理并在 3 个时间单元之后准备解码。分组 3 被解调引擎 1 和 2 处理并在 4 个时间单元之后准备解码。因此，分组 2、1 和 3 的解调延迟分别为 2 个、3 个和 4 个时间单元。

[0056] 表 1 总结了对于上面描述的 3 个解调资源分配方案的分组 1、2 和 3 的解调延迟。

如表 1 所示,将解调资源分配给片的方式影响到在片中发送的分组的解调延迟。

[0057] 表 1- 解调延迟

[0058]

分组	方案 1	方案 2	方案 3	单位
1	4	2	3	时间单元
2	4	4	2	时间单元
3	3	1	4	时间单元

[0059] 一旦可以从解调引擎 1-4 获得子分组,就可以分配解码引擎 1 和 2 来解码子分组。一旦分组的所有子分组均被解码,就可以分配分组引擎来处理每一个分组。在下面的实施例中,对于每一个子分组的解码时间等于该子分组的长度。对于每一个分组的分组处理时间等于该分组的长度。子分组 1 和 2 具有 2.0 个时间单元的长度。子分组 3、4 和 5 具有 2.5 个时间单元的长度。子分组 6 具有 1.5 个时间单元的长度。

[0060] 图 7 示出了使用第 1 个解调资源分配方案的对分组 1、2 和 3 进行解码和分组处理的实例。如上述图 6B 所示,子分组 6 在时刻 3 可用于解码,子分组 1-5 在时刻 4 可用于解码。子分组 6、2 和 4 被分配给解码引擎 1,子分组 1、3 和 5 被分配给解码引擎 2。解码引擎 1 从时刻 3 到时刻 4.5 对子分组 6 进行解码,从时刻 4.5 到时刻 6.5 对子分组 2 进行解码,从时刻 6.5 到时刻 9 对子分组 4 进行解码。解码引擎 2 从时刻 4 到时刻 6 对子分组 1 进行解码,从时刻 6 到时刻 8.5 对子分组 3 进行解码,从时刻 8.5 到时刻 11 对子分组 5 进行解码。

[0061] 仅具有子分组 6 的分组 3 在时刻 4.5 可用于分组处理。具有 2 个子分组 1 和 2 的分组 1 在时刻 6.5 可用于分组处理。具有 3 个子分组 3、4 和 5 的分组 2 在时刻 11 可用于分组处理。分组引擎从时刻 4.5 到时刻 6 处理分组 6,从时刻 6.5 到时刻 10.5 处理分组 1,从时刻 11 到时刻 18.5 处理分组 2。分组 3、1 和 2 的总的延迟分别为 6 个、10.5 个和 18.5 个时间单元。

[0062] 图 8 示出了使用第 2 个解调资源分配方案的对分组 1、2 和 3 进行解码和分组处理的实例。如上述图 6C 所示,子分组 6 在时刻 1 可用于解码,子分组 1 和 2 在时刻 2 可用于解码,子分组 3、4 和 5 在时刻 4 可用于解码。子分组 6、2 和 4 被分配给解码引擎 1,子分组 1、3 和 5 被分配给解码引擎 2。解码引擎 1 从时刻 1 到时刻 2.5 对子分组 6 进行解码,从时刻 2.5 到时刻 4.5 对子分组 2 进行解码,从时刻 4.5 到时刻 7 对子分组 4 进行解码。解码引擎 2 从时刻 2 到时刻 4 对子分组 1 进行解码,从时刻 4 到时刻 6.5 对子分组 3 进行解码,从时刻 6.5 到时刻 9 对子分组 5 进行解码。

[0063] 仅具有子分组 6 的分组 3 在时刻 2.5 可用于分组处理。具有 2 个子分组 1 和 2 的分组 1 在时刻 4.5 可用于分组处理。具有 3 个子分组 3、4 和 5 的分组 2 在时刻 9 可用于分组处理。分组引擎从时刻 2.5 到时刻 4 处理分组 6,从时刻 4.5 到时刻 8.5 处理分组 1,从时刻 9 到时刻 16.5 处理分组 2。因此,分组 3、1 和 2 的总的延迟分别为 4 个、8.5 个和 16.5 个时间单元。

[0064] 图 9 示出了使用第 3 个解调资源分配方案的对分组 1、2 和 3 进行解码和分组处理

的实例。如上述图 6D 所示,子分组 3、4 和 5 在时刻 2 可用于解码,子分组 1 和 2 在时刻 3 可用于解码,子分组 6 在时刻 4 可用于解码。子分组 3、5 和 6 被分配给解码引擎 1,子分组 4、1 和 2 被分配给解码引擎 2。解码引擎 1 从时刻 2 到时刻 4.5 对子分组 3 进行解码,从时刻 4.5 到时刻 7 对子分组 5 进行解码,从时刻 7 到时刻 8.5 对子分组 6 进行解码。解码引擎 2 从时刻 2 到时刻 4.5 对子分组 4 进行解码,从时刻 4.5 到时刻 6.5 对子分组 1 进行解码,从时刻 6.5 到时刻 8.5 对子分组 2 进行解码。

[0065] 具有 3 个子分组 3、4 和 5 的分组 2 在时刻 7 可用于分组处理。具有 2 个子分组 1 和 2 的分组 1 在时刻 8.5 可用于分组处理。仅具有子分组 6 的分组 3 在时刻 8.5 可用于分组处理。分组引擎从时刻 7 到时刻 14.5 处理分组 2,从时刻 14.5 到时刻 18.5 处理分组 1,从时刻 18.5 到时刻 20 处理分组 3。分组 2、1 和 3 的总的延迟分别为 14.5 个、18.5 个和 20 个时间单元。

[0066] 图 10 示出了使用第 3 个解调资源分配方案的对分组 1、2 和 3 进行解码和分组处理的另一个实施例。在这个实施例中,子分组 1、3 和 5 被分配给解码引擎 1,子分组 2、6 和 4 被分配给解码引擎 2。解码引擎 1 从时刻 3 到时刻 5 对子分组 1 进行解码,从时刻 5 到时刻 7.5 对子分组 3 进行解码,从时刻 7.5 到时刻 10 对子分组 5 进行解码。解码引擎 2 从时刻 3 到时刻 5 对子分组 2 进行解码,从时刻 5 到时刻 6.5 对子分组 6 进行解码,从时刻 6.5 到时刻 9 对子分组 4 进行解码。分组引擎从时刻 5 到时刻 9 处理分组 1,从时刻 9 到时刻 10.5 处理分组 3,从时刻 10.5 到时刻 18 处理分组 2。分组 1、3 和 2 的总的延迟分别为 9 个、10.5 个和 18 个时间单元。图 9 和 10 示出了通过改变解码资源的分配可以获得不同的总的延迟并且可能减少总的延迟。

[0067] 表 2 总结了对于图 7-10 中给出的 4 个实例的分组 1、2 和 3 的总的延迟。如表 2 所示,向分组分配解码资源的方式影响到分组的总的延迟。

[0068] 表 2- 总的延迟

[0069]

分组	图 7	图 8	图 9	图 10	单位
1	10.5	8.5	18.5	9	时间单元
2	18.5	16.5	14.5	18	时间单元
3	6	4	20	10.5	时间单元

[0070] 如图 6A-10 和表 2 所示,即使解码和分组处理的延迟被控制为最小,将 3 个分组分配到 4 个解调引擎和 2 个解码引擎的不同分配方式会导致这些分组的不同的总的延迟。可以看出,图 8 示出的实例具有为 16.5 个时间单元的最短的总的延迟,以及为 1、2 和 4 个时间单元的最短的解调延迟。相反,图 9 示出的实例具有 20 个时间单元的最长的总的延迟。

[0071] 通常,若考虑到所有情况时接收机处理器的吞吐量-延迟的性能界限可能不很清楚。由于并非所有的接收机的操作总是必须的,并且当省略不同的接收机操作时,其可能对吞吐量-延迟的性能造成不同程度的影响,这使得性能界限的不确定性可能会进一步复杂化。这意味着接收机处理器为满足特定的吞吐量-延迟的性能所要求的最小资源可能难以确定。这个问题的典型解决方案包括 (1) 逼近但不打算满足所宣称的性能, (2) 通过过度

预算接收机的资源来确保满足所宣称的性能，(3) 使用足够的内置支持设备来确保满足所宣称的性能。考虑多种原因所有这些方案都不是所期望的。

[0072] 在一个方面，可以动态管理接收机资源以允许具备任何配置和能力的接收机（例如，接收机处理器、宽带数据调制解调器等）在执行中逼近它的实际吞吐量 - 延迟的性能边界。动态接收机资源管理可以基于数据处理的需要动态地分配或指派可用的接收机资源（例如，以逐帧为基础）。动态接收机资源管理可以在尽可能小地影响吞吐量的情况下尝试限制和 / 或减少延迟，以满足服务质量 (QoS) 要求。

[0073] 图 11 示出了用于动态管理接收机资源的过程 1100 的设计方案。开始时，可以估计接收机要处理的至少一个分组中的每一个所需的接收机资源使用量（框 1112）。每一个分组所需的接收机资源使用量是用于处理该分组的接收机资源的数量。每一个分组所需的接收机资源使用量可以包括 (i) 解调资源使用量，其是用于对分组进行解调的解调资源的数量，(ii) 解码资源使用量，其是用于对分组进行解码的解码资源的数量，(iii) 分组处理资源使用量，和 / 或 (iv) 其他资源使用量。

[0074] 可以基于每一个分组的估计的接收机资源使用量和一个或多个分配标准将可用的接收机资源分配给至少一个分组（框 1114）。可以基于每一个分组的延迟要求、优先级和 / 或其他参数来分配可用的接收机资源。可以基于接收机资源分配来确定每一个分组的延迟（框 1116）。

[0075] 然后可以判断是否每一个分组的延迟满足该分组的延迟要求（框 1118）。如果答案为‘是’，则可以调度接收机按照接收机资源分配的指示来处理至少一个分组（框 1126）。否则，如果框 1118 的答案为‘否’，那么需要判断是否已经考虑了所有可能的分配方案（框 1120）。如果答案为“否”，则过程返回到框 1114，可以基于每一个分组的接收机资源使用量和一个或多个分配标准将可用的接收机资源分配给至少一个分组。可以按系统的方式执行框 1114-1120。框 1114-1120 可以尝试连续经历不同的接收机阶段以为每一个分组分配接收机资源，并累积这些接收机阶段的延迟，以及按照 QoS 要求来减少延迟。

[0076] 如果框 1120 的答案为‘是’，则可以跳过或推迟对一个或多个分组 / 子分组的接收机处理（例如，解调和 / 或解码），下面还会作进一步描述（框 1122）。随后可以判断是否每一个分组的延迟满足该分组的延迟要求（框 1124）。如果答案为‘是’，则可以调度接收机按照接收机资源分配的指示来处理至少一个分组（框 1126）。否则，如果框 1124 的答案为‘否’，则过程返回框 1122 以跳过或推迟一个或多个另外的分组 / 子分组、用其他方式对接收机的操作划分优先级等。框 1122 和 1124 可以被反复执行，直到要处理的每一个分组在它的延迟界限之内，可用的接收机资源被尽可能有效地分配，并且框 1124 获得‘是’的结果。

[0077] 在框 1112，可以基于诸如带宽的量和 / 或分组使用的时间量、分组使用的片的数目、发送分组所经过的层的数目、分组的调制阶数等多种因素来估计每一个分组的解调资源使用量。层是空间维度。对于 MIMO 传输，数据可以同时通过多个层发送。调制阶数与在每一个调制符号中所发送的比特数目有关。通常，更多的解调资源可以用于更多的带宽、更长的时间分配、更多的片、更多的层、以及更高的调制阶数。

[0078] 可以基于诸如分组的长度或大小、分组中子分组的数目以及子分组的长度、对分组执行的解码迭代次数、用于分组的编码方案等多个因素来估计每一个分组的解码资源使

用量。通常,更多的解码资源可以用于更长的分组长度和更多的迭代。

[0079] 在框 1114,可以根据上面描述的 3 个解调资源分配方案中的任意一个或某个其他的解调资源分配方案将可用的解调资源分配给至少一个分组。可用的解调资源可以包括预定数目的可并行操作的解调引擎和 / 或其他处理单元。可以根据任意解码资源管理方案将可用的解码资源分配给至少一个分组。可用的解码资源可以包括预定数目的可并行操作的解码引擎和 / 或其他处理单元。

[0080] 通常,可以根据一个或多个标准来分配可用的接收机资源。在一个设计方案中,可以按照基于每一个分组的优先级确定的连续顺序将可用的接收机资源分配给至少一个分组。具有最高优先级的分组可以被首先分配接收机资源,具有最低优先级的分组可以被最后分配接收机资源。在另一种设计方案中,可以按照基于延迟要求确定的连续顺序将可用的接收机资源分配给至少一个分组。具有最短延迟要求或最大延迟敏感性的分组(例如,用于基于 IP 的语音技术 (VoIP) 的分组)可以被首先分配接收机资源,具有最长延迟要求或最小延迟敏感性的分组(例如,用于网络冲浪的分组)可以被最后分配接收机资源。这些设计方案可以减轻由于有限的接收机资源数量引起的性能降低。

[0081] 在一个设计方案中,可以根据正确解码每一个子分组的可能性将可用的解码资源分配给子分组。可以发送每一个子分组以使得对于目标次数的 HARQ 传输来说能够以目标成功概率解码该子分组。正确解码子分组的可能性与针对该子分组所接收到的 HARQ 传输的次数有关。可以根据针对每一个子分组所接收到的 HARQ 传输的次数给子分组分配解码资源。例如,具有最多的 HARQ 传输次数的子分组可以被首先选择来解码,具有最少的 HARQ 传输次数的子分组可以被最后选择来解码。这样可以确保在所有子分组的解码资源使用量超出了可用的解码资源时,对可使用解码资源的有效使用。

[0082] 在一个设计方案中,可以根据每一个分组中子分组的数目和 / 或分组的大小来选择进行解码的分组。具有最少的子分组数目和 / 或最短长度的分组可以被最后选择来解码,具有最多的子分组数目和 / 或最长长度的分组可以被首先选择来解码。还可以按照从最小的分组到最大的分组的顺序选择分组进行解码。这种设计方案可以降低分组的延迟,提高分组的吞吐量,和 / 或提供其他好处。

[0083] 在一个设计方案中,可以根据每一个分组所使用的片的数目将可用的解调资源分配给分组。在一个设计方案中,具有最少数目片的分组可以被最后选择进行解调,具有最多数目片的分组可以被首先选择进行解调。还可以按照片的数目从最小到最大的顺序选择分组进行解调。在另一个设计方案中,可以选择分组所使用的 K_1 个片中的一个片而不是分组所使用的 K_2 个片中的一个片来进行解调,其中 $K_1 > K_2$ 。这些设计方案可以降低分组的延迟,提高分组的吞吐量,和 / 或提供其他好处。

[0084] 通常,为了减少延迟和提高吞吐量,符合任何跨层的 QoS 约束,可以以任何方式为分组分配解调资源和解码资源。在特定的方案中,如果解码是成功的,首先对较小的分组进行解调和解码可能产生更短的延迟。对于 HARQ,首先对较大的分组进行解调和解码可以释放更多的无线电资源以用于下一帧中的传输和处理。可以执行计算机仿真和 / 或其他分析来确定在不同的方案中对于不同的资源分配方案的延迟和吞吐量性能。对于每一个方案,可以选择使用能够提供最好性能的资源分配方案。

[0085] 对于框 1122,一个或多个分组 / 子分组的接收机处理可以被跳过或推迟,使得每

一个要处理的分组能够满足它的 QoS 要求。可以以多种方式来选择被跳过或推迟的分组。

[0086] 在一个设计方案中,可以基于所有要被处理的分组 / 子分组的延迟要求和 / 或优先级,来跳过或推迟对一个或多个分组 / 子分组的解码。具有最低延迟敏感性和 / 或最低优先级的分组 (例如,用于网络冲浪的分组) 可能被跳过或推迟,而不是具有最高延迟敏感性和 / 或最高优先级的分组 (例如,用于 VoIP 的分组)。

[0087] 在另一个设计方案中,可以基于正确解码每一个子分组的可能性,来跳过或推迟对一个或多个分组 / 子分组的解码。对于 HARQ,可以基于针对每一个子分组所接收到的 HARQ 传输的次数来跳过或推迟子分组。例如,可以跳过或推迟具有最少 HARQ 传输次数的子分组。

[0088] 在另一个设计方案中,可以基于每一个分组的子分组数目和 / 或分组的大小,来跳过或推迟对一个或多个分组 / 子分组的解码。例如,可以跳过或推迟具有 L1 个子分组的分组中的一个分组,而不是具有 L2 个子分组的分组中的一个分组,其中 $L1 < L2$ 。

[0089] 在一个设计方案中,可以基于每一个分组所使用的片的数目,来跳过或推迟对一个或多个片的解调。例如,可以跳过或推迟采用最多数目的片进行发送的分组。在另一个实例中,可以跳过或推迟用于一个分组的 N1 个片中的一个片,而不是用于一个分组的 N2 个片中的一个片,其中 $N1 > N2$ 。

[0090] 也可以根据其他标准来跳过或推迟对一个或多个分组 / 子分组的接收机处理。通常,诸如延迟要求、优先级、解码概率、HARQ 传输的次数、每个分组中子分组的数目、分组的大小、以及每个分组中片的数目之类的标准可用于在框 1114 中将可用的接收机资源分配给分组,和 / 或在框 1122 中选择一个或多个分组 / 子分组以跳过或推迟接收机处理。在一个设计方案中,可以每次按照顺序将可用的接收机资源分配给一个分组,该顺序可以根据前面给出的标准中的任何一个或多个来确定。

[0091] 图 12 描述了用于动态管理接收机资源的装置 1200 的设计方案。装置 1200 包括用于估计要被接收机处理的至少一个分组中的每一个所需的接收机资源使用量的模块 (模块 1212),用于根据每个分组的估计的接收机资源使用量以及一个或多个分配标准将可用的接收机资源分配给至少一个分组的模块 (模块 1214),用于根据接收机资源分配确定每个分组的延迟的模块 (模块 1216),如果需要,用于跳过或推迟对一个或多个分组 / 子分组的接收机处理的模块 (模块 1218),以及用于按照接收机资源分配的指示调度接收机来处理至少一个分组的模块 (模块 1220)。

[0092] 图 12 中的模块可以包括处理器、电子设备、硬件设备、电子部件、逻辑电路、存储器等,或它们的任意组合。

[0093] 图 13 示出了可用于图 4 中接收机处理器 440 和 460 中每一个的接收机处理器 502 的设计框图。接收机处理器 502 包括图 5 中接收机处理器 500 中的所用引擎和复用器。接收机处理器 502 进一步包括动态接收机资源管理器 540。管理器 540 可以接收诸如带宽分配信息 (例如,分配的片)、数据分段信息 (例如,在分配的片中发送的子分组和分组)、QoS 要求 (例如,延迟要求)、和 / 或其他信息 (例如,每个分组的优先级) 的输入。管理器 540 可以根据所接收的信息并依照上面描述的任意方案为需要处理的分组分配或指定可用的接收机资源。管理器 540 可以实现图 11 中的过程 1100 和 / 或用于向分组动态分配可用的接收机资源的其他过程。

[0094] 接收机资源的动态管理可以提供如下优点：

[0095] ●当可用的接收机资源能够支持所期望的吞吐量时，获得来自接收机最小可能的延迟，

[0096] ●另外，在具有最大可能的吞吐量时保持来自接收机的 QoS 界限延迟，

[0097] ●在尽可能少的浪费的情况下确保可用的接收机资源的高效利用，以及

[0098] ●允许跨层 QoS 延伸至物理层处理。

[0099] 动态接收机资源管理可以允许对可用的接收机资源的高效分配和使用，以获得高吞吐量的和低延迟。动态接收机资源管理可以确保所有使用被正确地处理以获得最好的实际吞吐量 - 延迟性能，而无需知道绝对的最坏情况操作方案。动态接收机资源管理可以获得更高的吞吐量和 / 或更低的延迟，从而满足包括更多传统应用的很宽范围应用的 QoS 要求，其中的应用例如低带宽低延迟的 VoIP，高带宽中等延迟的视频点播 (VoD)，高带宽低延迟的视频电话 (VT)，任意带宽高延迟的网络冲浪，以及宽带接收机的高带宽的可用性所带来的未来将会出现的应用。动态接收机资源管理对于所有的操作方案和同时发生的应用的任意混合可以允许有效地利用可用的接收机资源。动态接收机资源管理可以被应用于基于能够处理多数据流和 / 或多用户操作的多阶段、多线程和 / 或多处理器架构的任何无线或有线接收机设计和实现。

[0100] 可以通过多种手段实现本文描述的技术。例如，这些技术可以由硬件、固件、软件、或它们的结合实现。在硬件实现中，用于执行该技术的处理单元（例如，图 13 中的接收机处理器 502）可以在一个或多个专用集成电路 (ASIC)、数字信号处理器 (DSP)、数字信号处理设备 (DSPD)、可编程逻辑器件 (PLD)、现场可编程门阵列 (FPGA)、处理器、控制器、微控制器、微处理器、电子设备、设计用来执行本文描述的功能的其他电子单元、计算机、以及它们的组合中来实现。

[0101] 对于固件和 / 或软件实现，该技术可以由执行本文描述的功能的代码（例如，程序、函数、模块、指令等）来实现。通常，任何计算机 / 处理器可读介质有形地体现了用于实现本文描述的技术的固件和 / 或软件代码。例如，固件和 / 或软件代码可以被存储在存储器（例如，图 4 中的存储器 432 和 472）并被处理器（例如，处理器 430 和 470）执行。存储器可实施在处理器内部或处理器外部。固件和 / 或软件代码可以被存储在诸如随机存取存储器 (RAM)、只读存储器 (ROM)、非易失性随机存取存储器 (NVRAM)、可编程只读存储器 (PROM)、电可擦除 PROM (EEPROM)、闪存、软磁盘、压缩光盘 (CD)、数字多用途光盘 (DVD)、磁或光数据存储设备等的计算机 / 处理器可读介质中。代码可以被一个或多个计算机 / 处理器执行并可以促使计算机 / 处理器执行本文描述的功能性的某些方面。

[0102] 提供了本发明前面的描述以使得任何本领域技术人员能够实施和使用本发明。对本领域技术人员来说对本发明任何形式的修改都是显而易见的，同时在不偏离本发明的精神和范围的情况下可以将本文中定义的基本原则应用于其他变型。因此，本发明并不意图局限于本文描述的实例和设计方案，而是要与符合本文公开的原理和新颖性特征的最宽的范围相一致。

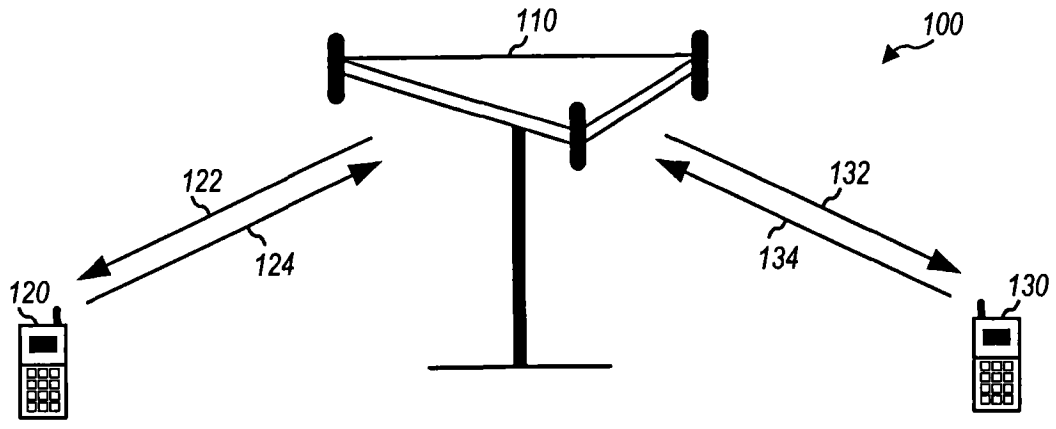


图 1

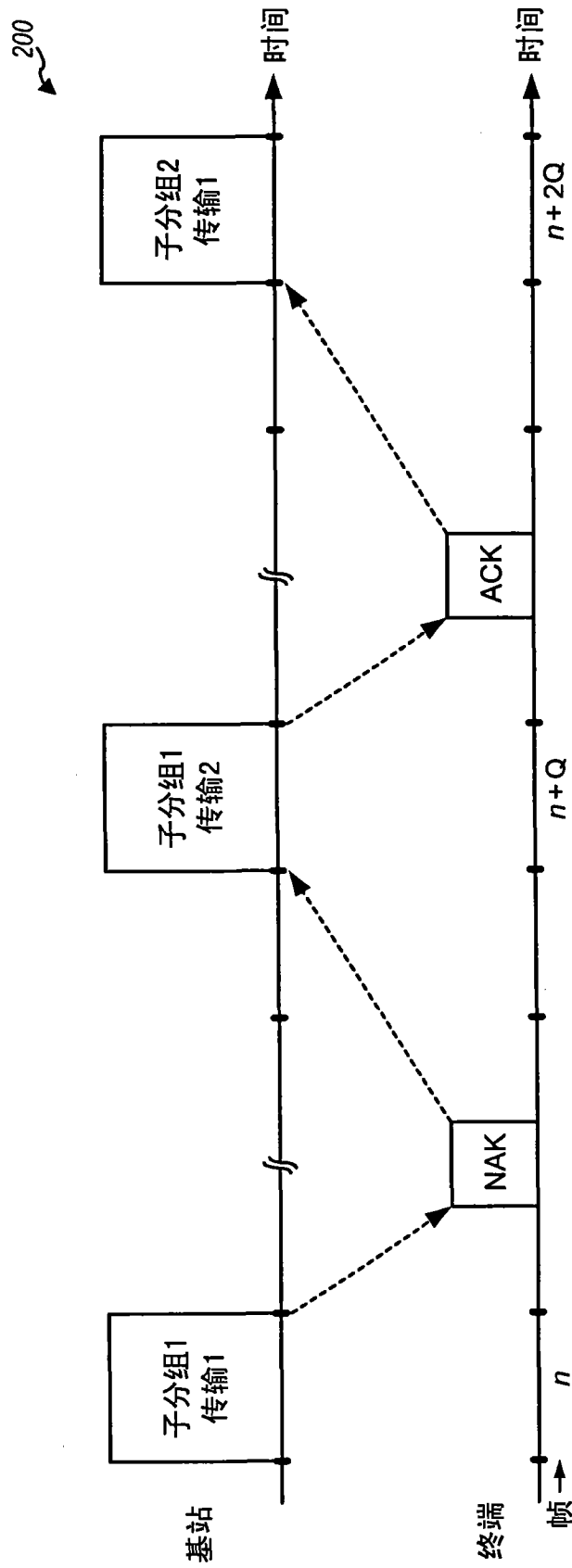


图 2

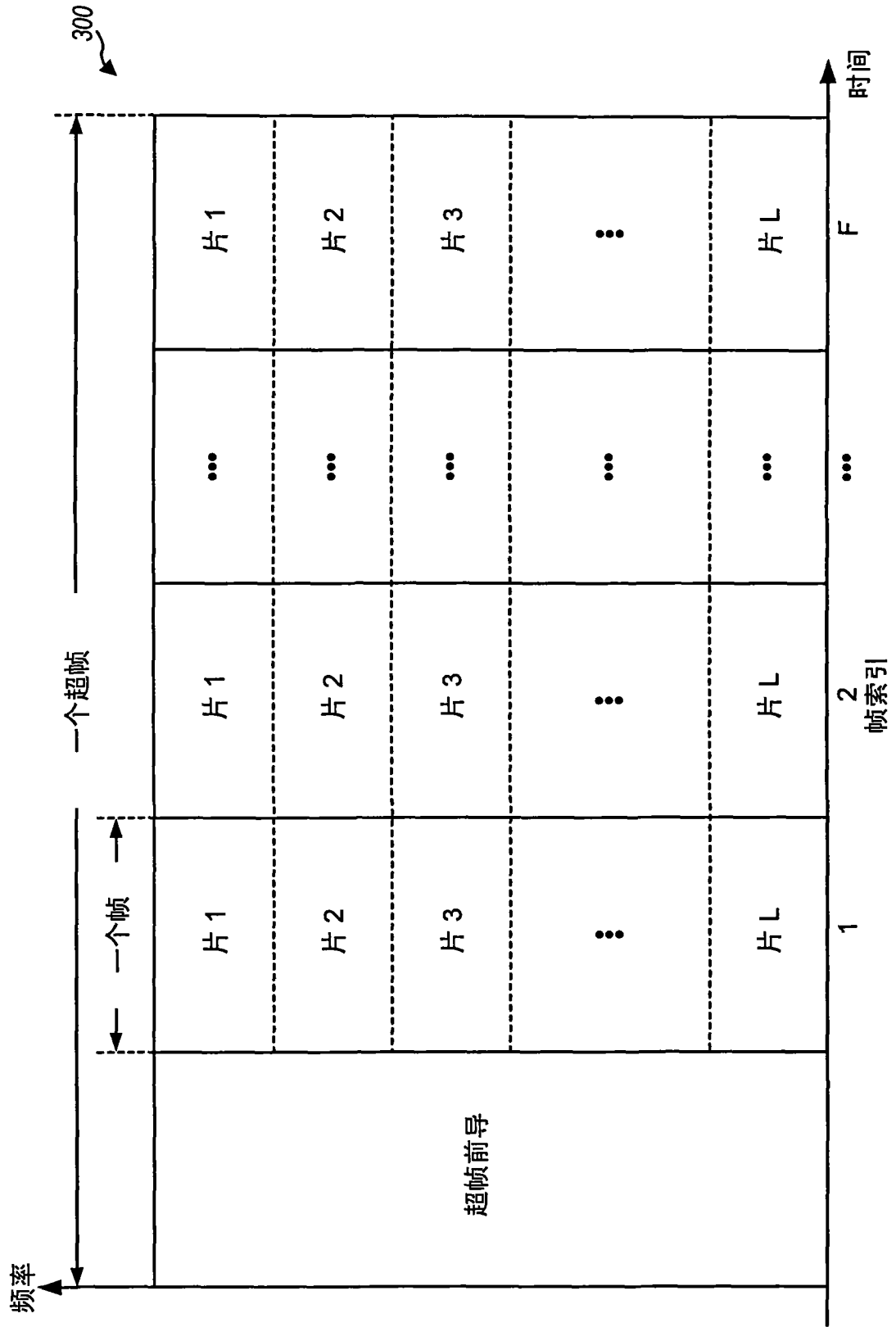


图 3

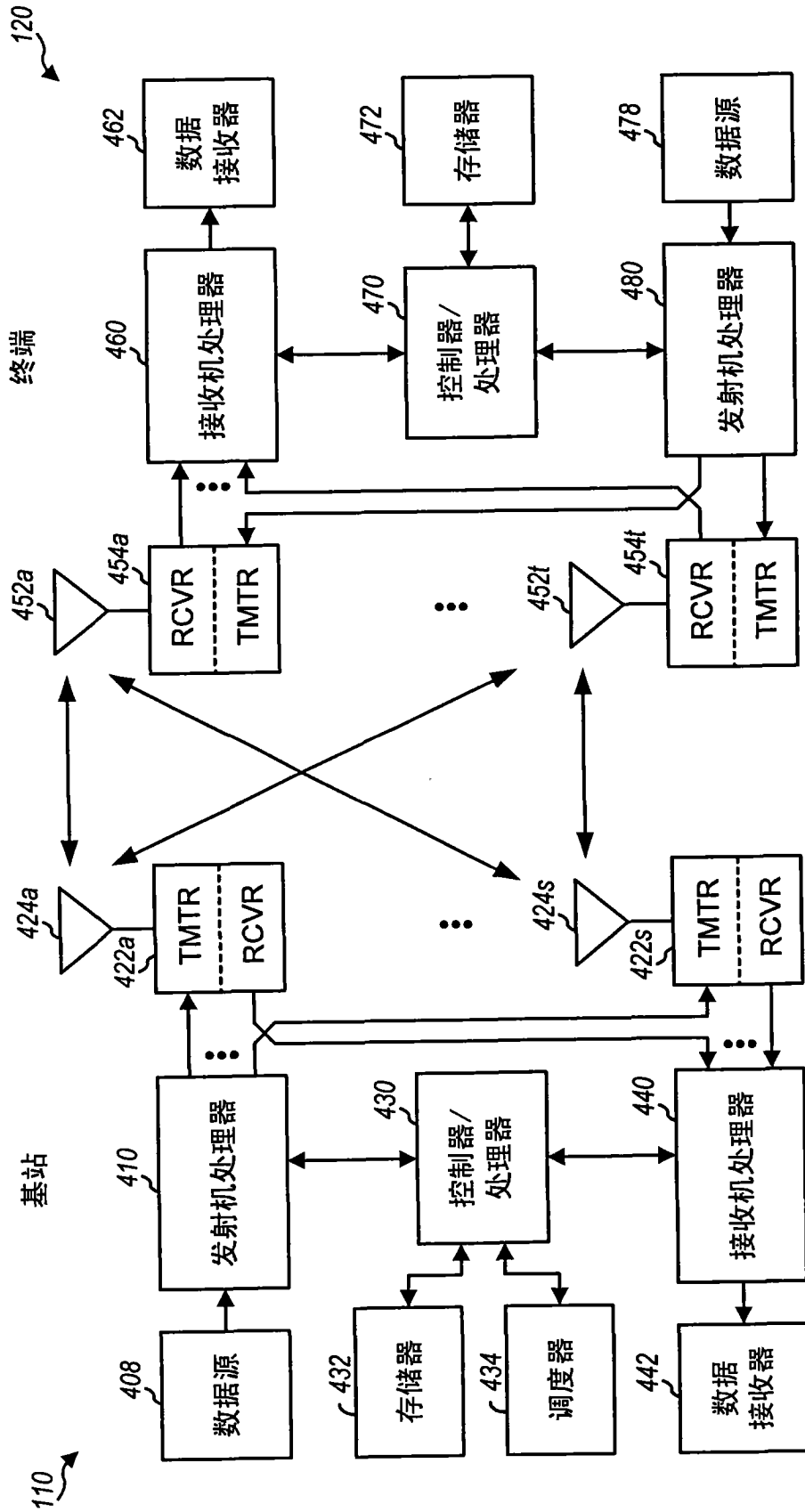


图 4

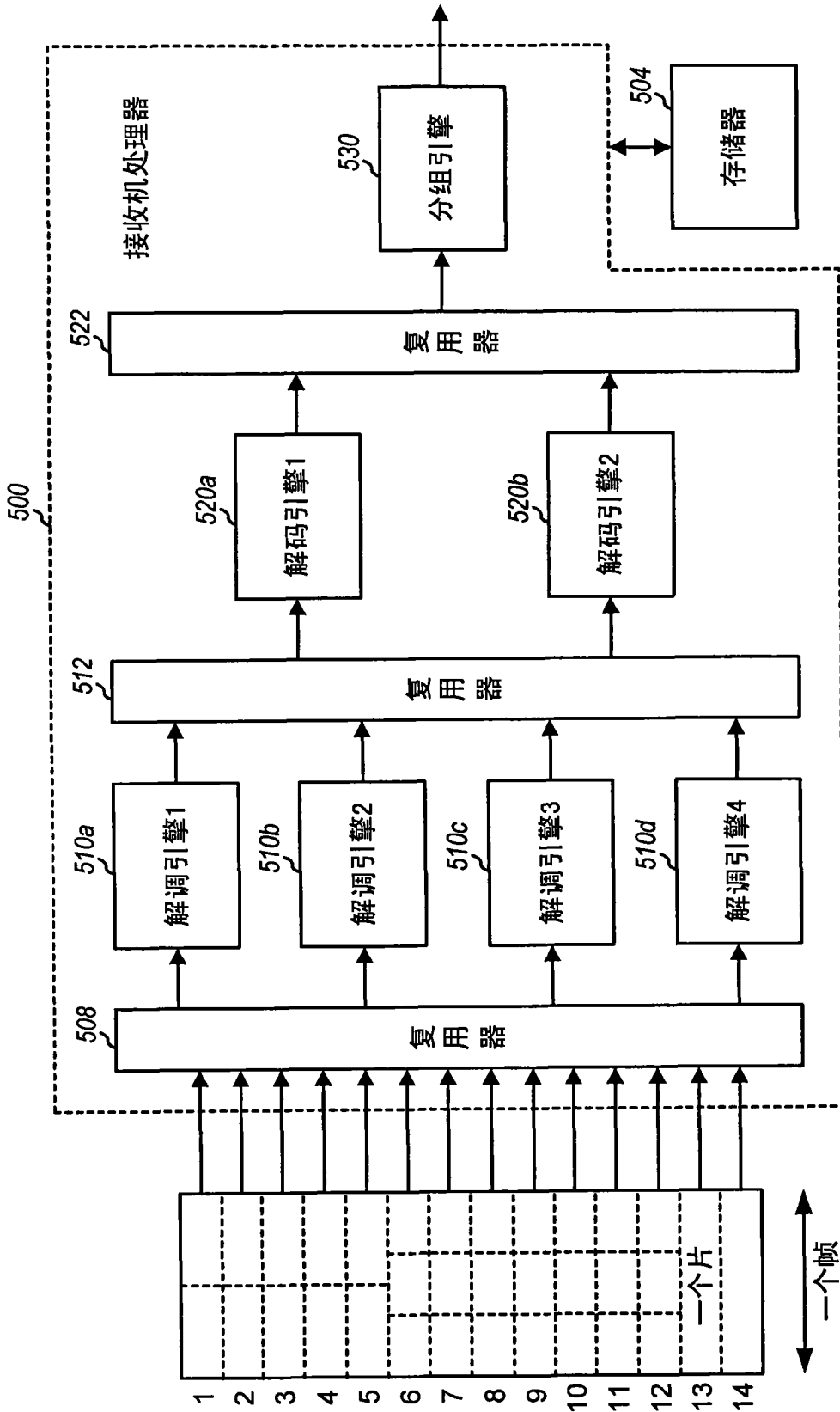


图 5

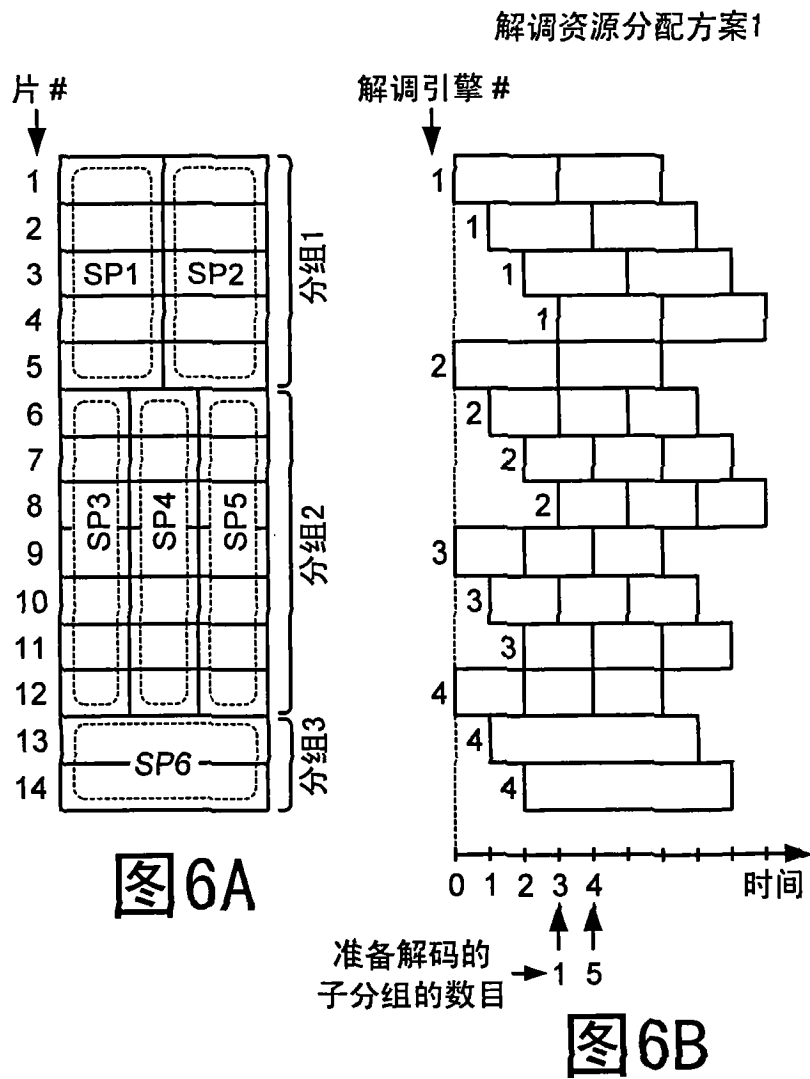


图 6A 图 6B

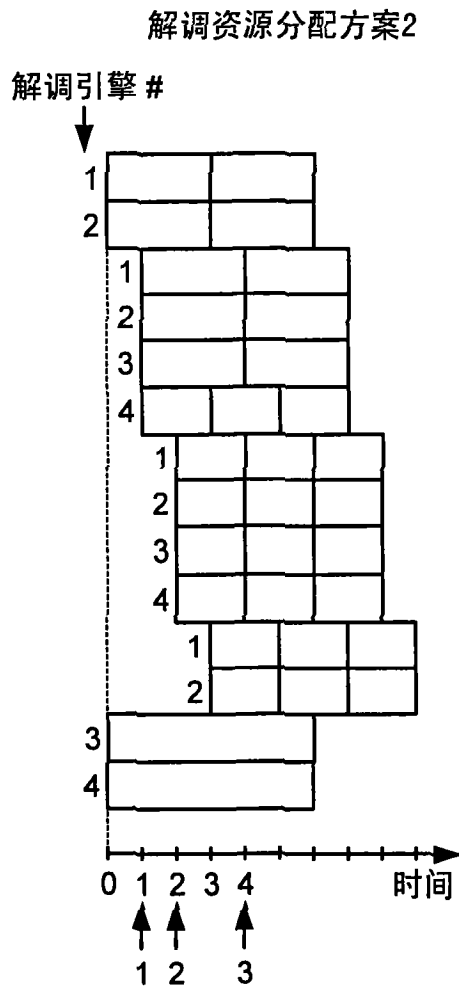


图 6C

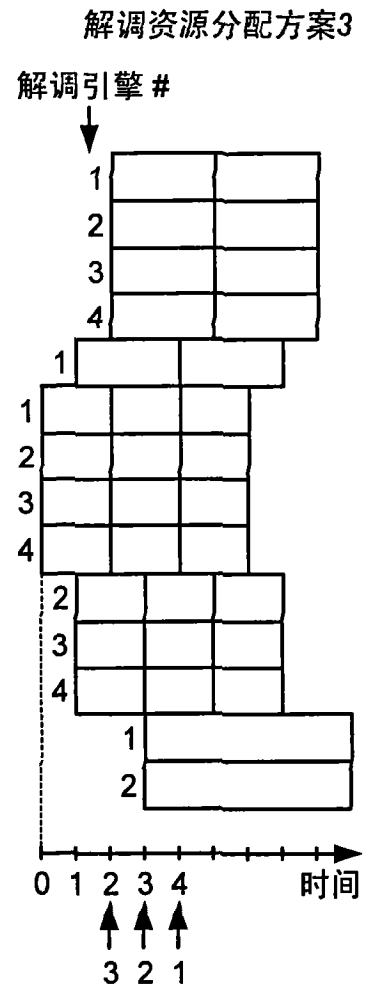


图 6D

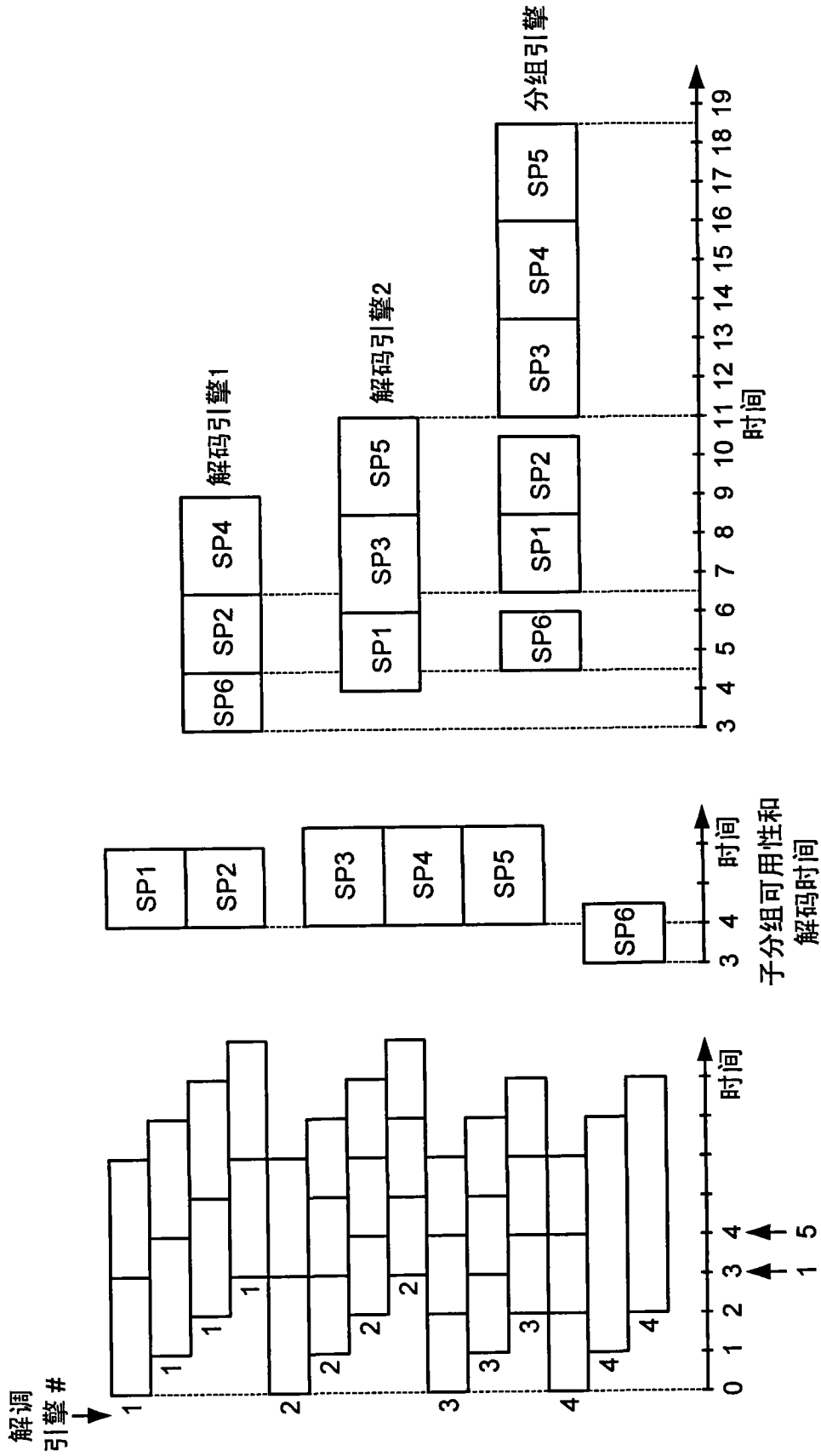


图 7

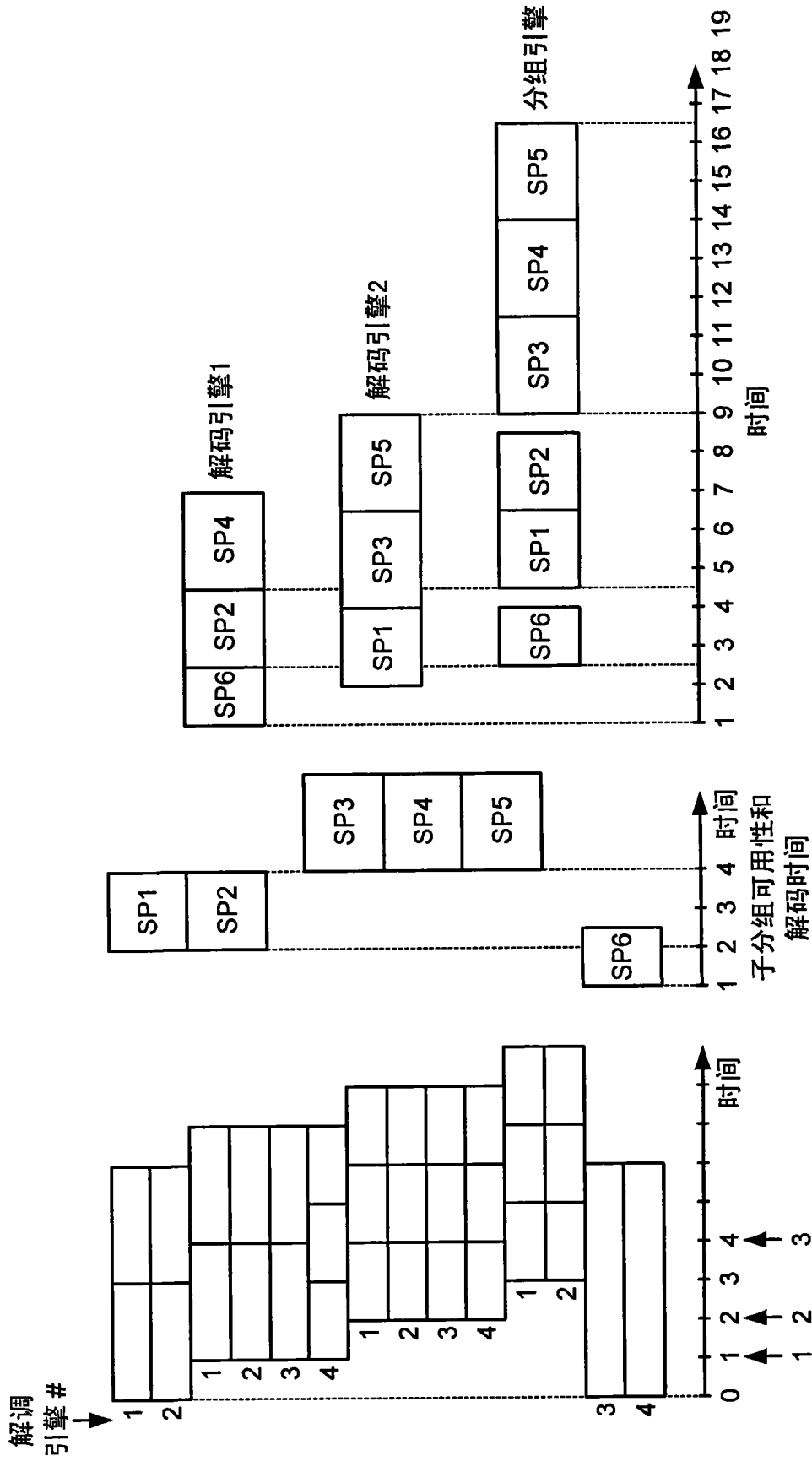


图 8

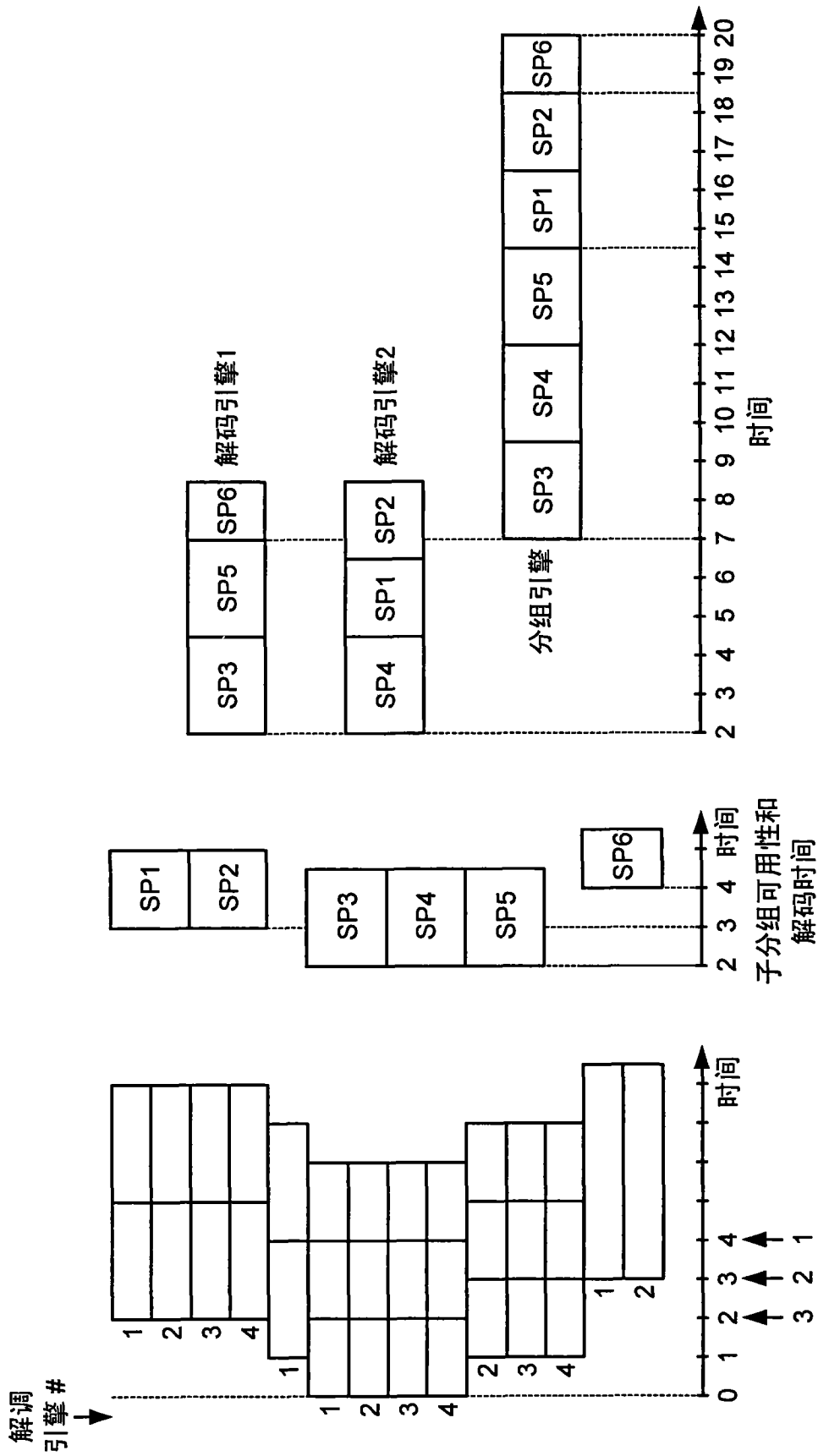


图 9

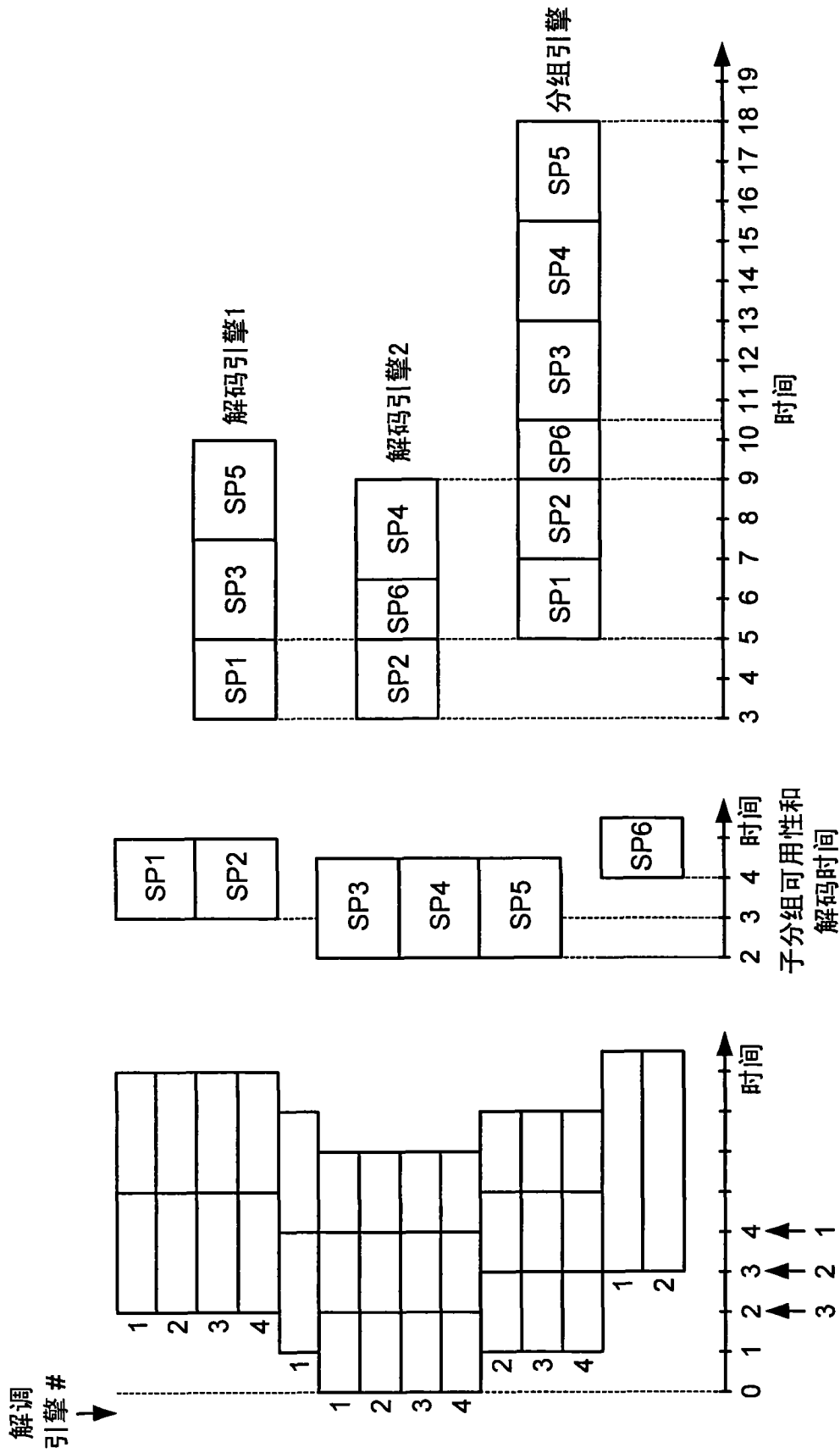


图 10

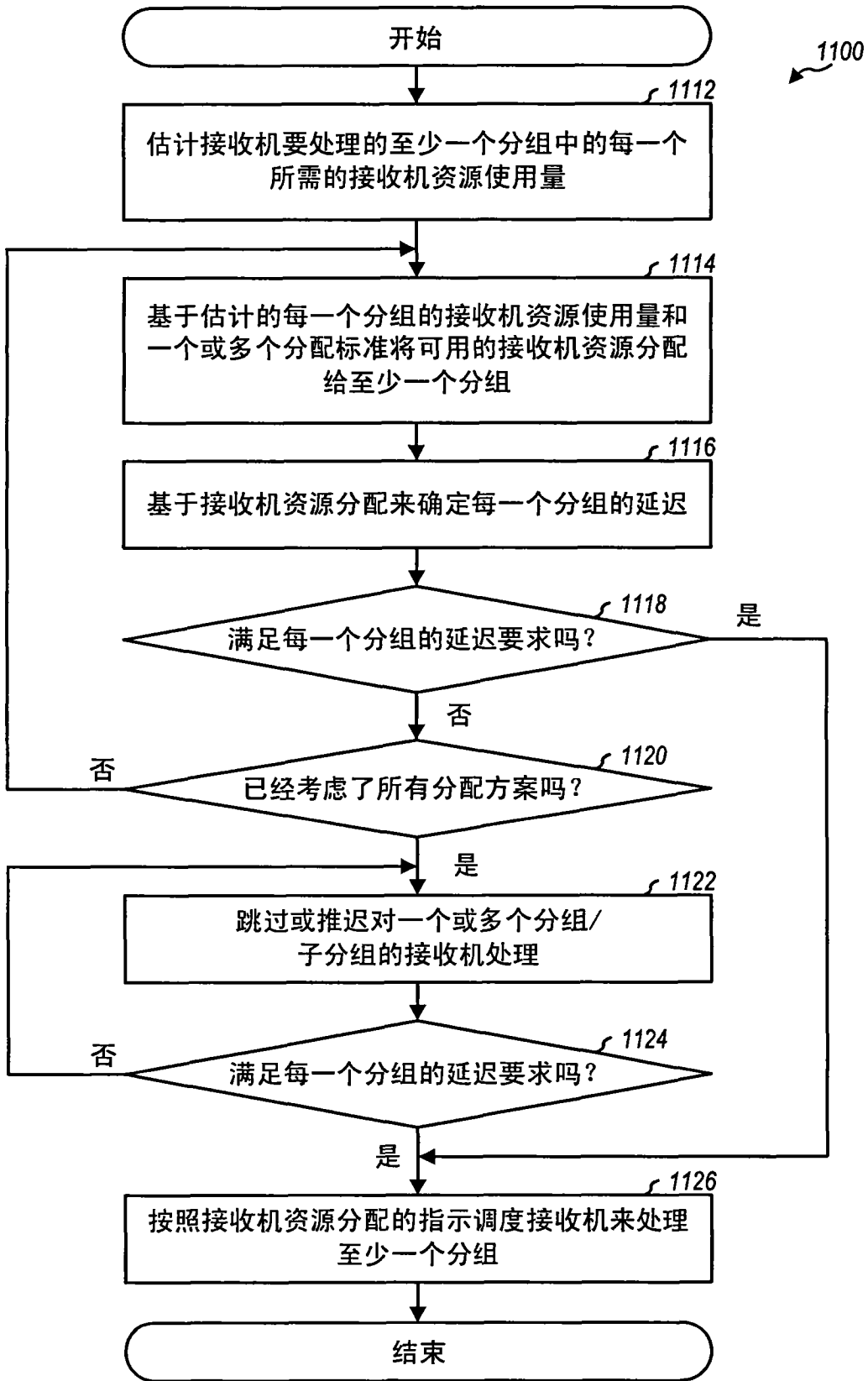


图 11

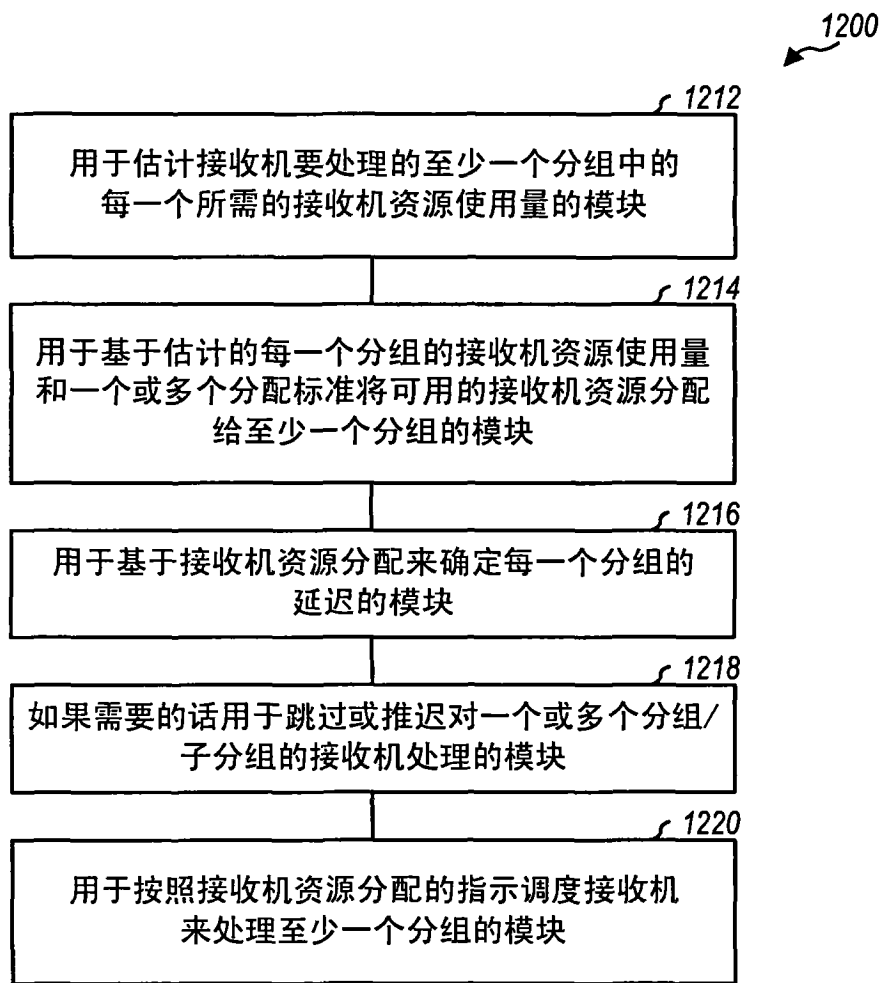


图 12

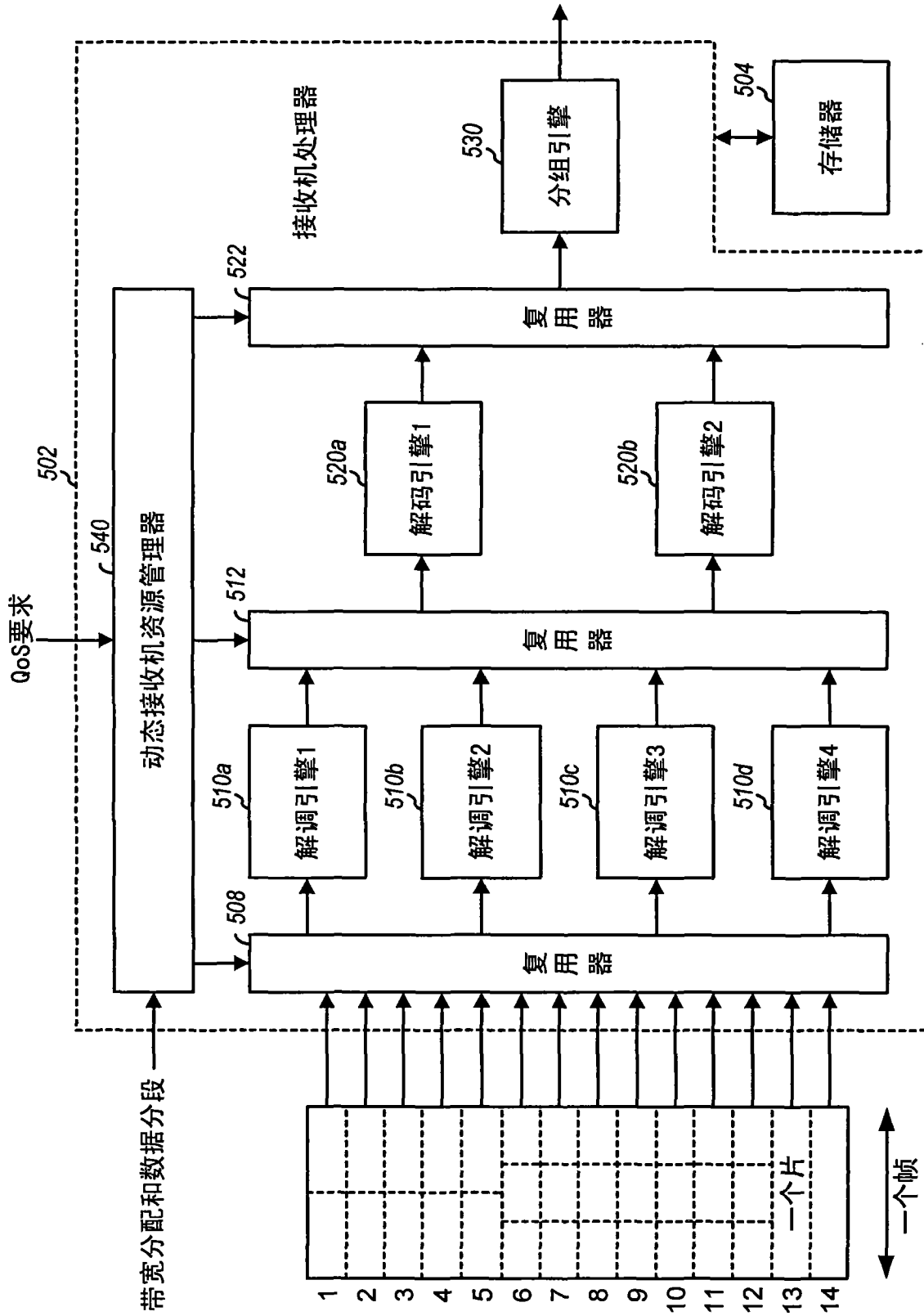


图 13