



## (12) 发明专利

(10) 授权公告号 CN 101027717 B

(45) 授权公告日 2011.09.07

(21) 申请号 200580013444.8

H03M 7/40 (2006.01)

(22) 申请日 2005.03.21

G10L 19/02 (2006.01)

## (30) 优先权数据

60/556,183 2004.03.25 US

## (56) 对比文件

10/911,062 2004.08.04 US

US 006385571 B1, 2002.05.07, 全文.

10/911,067 2004.08.04 US

US 006226616 B1, 2001.05.01, 全文.

## (85) PCT申请进入国家阶段日

EP 0869620 A2, 1998.10.07, 全文.

2006.10.27

US 20030179938 A1, 2003.09.25, 全文.

## (86) PCT申请的申请数据

CN 1253418 A, 2000.05.17, 全文.

PCT/US2005/009275 2005.03.21

US 006226608 B1, 2001.05.01, 说明书第8

## (87) PCT申请的公布数据

栏 33-36 行, 图 2.

W02005/098823 EN 2005.10.20

W0 03/077235 A1, 2003.09.18, 说明书第9

## (73) 专利权人 DTS(BVI) 有限公司

页 24-30 行, 第 11 页 4-9 行, 21-25 行, 第 12 页

地址 英属维尔京群岛

18-21 行, 图 3.

## (72) 发明人 左兰·菲左

CN 1239368 A, 1999.12.22, 全文.

## (74) 专利代理机构 中国国际贸易促进委员会专

US 006675148 B2, 2004.01.06, 说明书第2

利商标事务所 11038

栏 50-65 行.

代理人 马浩

审查员 辛杰

## (51) Int. Cl.

G10L 19/00 (2006.01)

权利要求书 4 页 说明书 14 页 附图 12 页

## (54) 发明名称

无损多通道音频编解码器

## (57) 摘要

一种无损音频编解码器将每帧内的音频数据分段以便在每个片段必须完全可解码并小于最大尺寸的约束下提高压缩性能。对于每个帧，编解码器选择片段持续时间和编码参数，例如，用于每个片段的特定熵编码器和其参数，其在上述约束下将整个帧的编码净荷降至最小。可以为每个通道选择不同的编码参数集或者为所有通道选择全局编码参数集。还可以通过为 M 通道音频形成 M/2 解相关通道来进一步提高压缩性能。通道的三重组（基准、相关、解相关）提供了两种可能的配对组合：基准、相关；和基准、解相关，其可以在分段和熵编码优化期间被考虑以进一步提高压缩性能。



1. 一种无损地编码多通道音频的方法,包括:  
将所述多通道音频分块成相等持续时间的帧;  
将每个帧分段成多个片段;  
在每个片段必须完全可解码并且具有小于最大字节数的编码片段净荷的约束下,为每个片段确定持续时间和熵编码参数以减小帧的尺寸可变的编码净荷;  
根据所述熵编码参数对所述帧内的每个通道熵编码所述片段;以及  
将用于每个片段的编码后的音频数据和所述熵编码参数打包到所述帧中;  
其中通过以下步骤确定片段的持续时间:
  - a) 将所述帧分割成给定持续时间的多个片段;
  - b) 为每个片段确定在每个通道中的编码参数集和编码净荷;
  - c) 为每个片段计算在所有通道上的编码净荷;
  - d) 如果任一片段在所有通道上的编码净荷超过最大尺寸,则丢弃所述编码参数集;
  - e) 如果对于当前分割的帧的编码净荷低于对于先前分割的最小编码净荷,则存储当前编码参数集并更新最小编码净荷;以及
  - f) 对不同持续时间的多个片段重复步骤 a 至 e。
2. 根据权利要求 1 的方法,其中所述片段持续时间最初被设置在最小持续时间并在每次分割迭代时增加。
3. 根据权利要求 2 的方法,其中所述片段持续时间最初被设置为二的幂并且在每次分割迭代时加倍。
4. 根据权利要求 2 的方法,其中如果任一片段在所有通道上的编码净荷超过所述最大尺寸,则该分割迭代终结。
5. 根据权利要求 1 的方法,其中所述编码参数集包括熵编码器及其参数的选择。
6. 根据权利要求 5 的方法,其中选择熵编码器及其参数以将该通道中该片段的编码净荷降至最小。
7. 根据权利要求 1 的方法,进一步包括为通道对生成解相关通道以便形成包含基准通道、相关通道和解相关通道的三重组,选择基准和相关通道对或基准和解相关通道对,并对所选择的通道对中的通道熵编码。
8. 根据权利要求 1 的方法,其中,基于哪个产生包括所述帧的头部和音频数据在内的最小编码净荷,所述确定的编码参数集或者是针对每个通道的不同的编码参数集或者是针对所有通道的全局性编码参数集。
9. 根据权利要求 1 的方法,其中所述片段的持续时间被确定为使得每帧的编码净荷降至最小。
10. 根据权利要求 1 的方法,其中所述片段的持续时间部分地是通过为每个片段选择包括多个熵编码器之一及其编码参数的编码参数集来确定的。
11. 根据权利要求 10 的方法,其中所述片段的持续时间部分地是通过为每个通道选择不同的编码参数集或为所有通道选择全局性编码参数集来确定的。
12. 根据权利要求 10 的方法,其中为不同的片段持续时间计算编码参数集,并且选择对应于具有满足每个片段必须完全可解码并且具有小于最大字节数的编码片段净荷的约束的最小编码净荷的编码参数集的持续时间。

13. 根据权利要求 1 的方法,进一步包括为通道对生成解相关通道以形成至少一个包含基准通道、相关通道和解相关通道的三重组,

所述片段的持续时间部分地是通过为每个所述三重组选择基准和相关通道对或基准和解相关通道对用于熵编码而确定的。

14. 根据权利要求 13 的方法,其中通过确定是解相关通道还是相关通道对编码净荷贡献最少比特来选择通道对。

15. 根据权利要求 13 的方法,其中两个最相关的通道形成第一对,如此类推,直到通道被耗尽;如果剩余奇数通道,则其形成基准通道。

16. 根据权利要求 15 的方法,其中在每对中,具有较小零滞后自相关估计的通道是基准通道。

17. 根据权利要求 16 的方法,其中通过将基准通道与解相关系数相乘并从相关通道减去该乘积结果来生成解相关通道。

18. 一种无损编码 PCM 音频数据的方法,包括:

处理多通道音频以创建包括基准通道和相关通道的通道对;

为每个通道对生成解相关通道以形成至少一个包含基准通道、相关通道和解相关通道的三重组;

将所述多通道音频分块成相等持续时间的帧;

在每个片段必须完全可解码并且小于最大尺寸的约束下,将每个帧分段成预定持续时间的多个片段、为每个片段确定编码参数并从所述至少一个三重组中选择基准和相关通道对或基准和解相关通道对,以将帧的编码净荷降至最小;

根据所述编码参数对所选通道对中每个通道的每个片段熵编码;以及

将所述编码后的音频数据打包到比特流中。

19. 根据权利要求 18 的方法,其中所述片段的预定持续时间部分地是通过选择多个熵编码器之一及其编码参数来确定的。

20. 根据权利要求 18 的方法,其中每个通道被分配包括所选择的熵编码器和其参数的编码参数集,所述片段的持续时间部分地是通过为每个通道选择不同的编码参数集或为所有通道选择全局性编码参数集来确定的。

21. 根据权利要求 18 的方法,其中对于帧中的每一个片段,所述预定持续时间相同。

22. 根据权利要求 18 的方法,其中为每个帧确定所述预定持续时间,并且该预定持续时间在整个帧序列内变化。

23. 一种多通道音频编码器,用于编码以已知采样率采样、具有音频带宽以及被分块成帧序列的数字音频信号,包括:

核心编码器,从数字音频信号提取核心信号并将其编码为核心比特;

打包器,将核心比特加上头部信息打包成第一比特流;

核心解码器,解码核心比特以形成重建的核心信号;

求和节点,对多音频通道中每个通道从所述重建的核心信号和数字音频信号形成差分信号;

无损编码器,将多通道差分信号的每帧分段成多个尺寸可变的片段,并将所述尺寸可变的片段熵编码成扩展比特,所述无损编码器在每个尺寸可变的片段被调整尺寸以便完全

可解码并且小于最大尺寸的约束下,选择片段持续时间以减小帧内差分信号的编码净荷;以及

打包器,将扩展比特打包成第二比特流;

其中该无损编码器通过以下步骤确定片段持续时间:

a) 将帧分割成给定持续时间的若干片段;

b) 为每个片段确定在每个通道中的编码参数集和编码净荷;

c) 为每个片段计算在所有通道上的编码净荷;

d) 如果任一片段在所有通道上的编码净荷超过最大尺寸,则丢弃该编码参数集;

e) 如果对于当前分割的帧的编码净荷小于对于先前分割的最小编码净荷,则存储当前编码参数集并更新所述最小编码净荷;以及

f) 对于不同持续时间的多个片段重复步骤 a 至 e。

24. 根据权利要求 23 的多通道音频编码器,其中核心编码器包括丢弃较高的 N/2 子带的 N 频带分析滤波器组,和仅编码较低的 N/2 子带的核心子带编码器,

该核心解码器包括将核心比特解码成较低的 N/2 子带的采样的核心子带解码器,和 N 频带合成滤波器组,该 N 频带合成滤波器组取出较低的 N/2 子带的采样,并使较高的 N/2 子带的未发送的子带采样归零,以及合成以已知采样率采样的重建音频信号。

25. 根据权利要求 23 的多通道音频编码器,其中无损编码器为通道对生成解相关通道以形成包含基准通道、相关通道和解相关通道的三重组,选择基准和相关通道对或基准和解相关通道对,并对所选通道对中的通道熵编码。

26. 根据权利要求 23 的多通道音频编码器,其中数字音频信号包括被组织成至少第一和第二通道集的多音频通道,所述第一通道集被核心编码器和无损编码器编码,所述第二通道集仅被所述无损编码器编码。

27. 根据权利要求 26 的多通道音频编码器,其中编码所述第一通道集的无损编码器包括 5.1 通道配置。

28. 根据权利要求 27 的多通道音频编码器,其中核心编码器具有编码核心信号的最大比特率。

29. 根据权利要求 26 的多通道音频编码器,其中核心编码器以预定采样率一半的采样率提取并编码核心信号。

30. 一种解码无损比特流的方法,包括:

接收作为帧序列的比特流,该帧序列包括含有片段数和每片段采样数的通用头部信息,用于每个通道集的含有消耗的字节、熵代码标记和编码参数的片段头部信息,以及存储在多个片段中的编码残留多通道音频信号;

对头部拆包以提取熵代码标记和编码参数以及编码的残留音频信号,并利用所选择的熵代码和编码参数对帧中每个片段执行熵解码,以便为每个片段生成残留音频信号;以及

对头部拆包以提取预测系数并对残留音频信号执行逆向预测,以便为每个片段生成 PCM 音频。

31. 根据权利要求 30 的方法,其中所述片段头部信息还包括所有通道相同参数标记,其指示熵代码和编码参数对于每个通道是否不同,或者它们对于所有通道是否相同。

32. 根据权利要求 30 的方法,其中比特流进一步包括含有成对通道解相关标记、原始

通道顺序、以及量化的通道解相关系数的通道集头部信息，所述逆向预测步骤生成解相关 PCM 音频，该方法进一步包括：

对该头部拆包以提取原始通道顺序、成对通道解相关标记和量化通道解相关系数，并执行逆向交叉通道解相关以生成多通道 PCM 音频。

33. 根据权利要求 32 的方法，其中成对通道解相关标记指示用于包含基准通道、相关通道和解相关通道的三重组的基准和相关通道对或基准和解相关通道对是否被编码，该方法进一步包括：

如果该标记指示基准和解相关通道对，则将相关通道与量化通道解相关系数相乘并将结果加到基准通道以生成相关通道。

## 无损多通道音频编解码器

[0001] 相关申请的交叉引用

[0002] 本申请要求 2004 年 3 月 25 日提交的名称为“后向兼容无损音频编解码器”的美国临时申请 No. 60/566,183, 在 35U. S. C 119(e) 下的优先权, 其全部内容在此并入作为参考。

### 技术领域

[0003] 本发明涉及无损音频编解码器, 尤其涉及具有改进的压缩性能的无损多通道音频编解码器。

### 背景技术

[0004] 当前许多低比特率有损音频编码系统用于消费者以及专业音频重放产品和业务的宽范围。例如, 杜比 AC3(杜比数字)音频编码系统是一种利用高达 640kbit/s 的比特率, 为激光盘、NTSC 编码的 DVD 视频和 ATV 编码立体声和 5.1 通道音频声道的世界范围的标准。MPEG I 和 MPEG II 音频编码标准被广泛用于比特率高达 768kbit/s 的 PAL 编码的 DVD 视频, 欧洲陆地数字无线电广播和美国卫星广播的立体声和多通道声道编码。DTS(数字影院系统)相关声学音频编码系统以高达 1536kbit/s 的比特率经常用于光盘、DVD 视频、欧洲卫星广播和激光盘的演播室质量 5.1 通道音频声道。

[0005] 最近, 许多消费者已经对这些所谓的“无损”编解码器表示兴趣。“无损”编解码器依靠压缩数据而不会丢弃任何信息的算法并产生与(数字化)的源信号相同的解码信号。这种性能的获得是以如下为代价的:这种编解码器典型地需要比有损编解码器更多的带宽, 并将该数据压缩到更少的程度。

[0006] 图 1 是无损地压缩单个音频通道所涉及的操作的框图表示。尽管多通道音频中的通道通常不是独立的, 这种依赖性通常很弱, 并且很难估计。因此, 各个通道典型地被分别压缩。然而, 某些编码器将试图通过形成简单的残留信号并编码(Ch1, Ch1-CH2)来消除相关性。更为复杂的方法采用, 例如, 通道尺寸上的几个连续的正交投影步骤。所有的技术都是基于首先从信号中去除冗余, 然后以有效的数字编码方案编码所产生的信号的原理。无损编码器包括 MPL(DVD Audio), Monkey's audio(计算机应用)、Apple lossless、Windows Media Prolossless、Audiopak、DVD、LTAC、MUSICcompress、OggSquish、Philips、Shorten、Sonarc 和 WA。对于多种这些编解码器的评论由 Mat Hans、Ronald Schafer 的“Lossless Compression of Digital Audio”Hewlett Packard, 1999 年提供。

[0007] 引入成帧 10 以便为可编辑性作准备, 数据的绝对量禁止重复解压要被编辑的区域之前的全部信号。音频信号被划分成相等持续时间的独立的帧。该持续时间不应太短, 因为可能从每个帧前缀的头部产生大量的开销。相反地, 帧持续时间不应该太长, 因为这将限制时间自适应性并使编辑更难。在许多应用中, 帧尺寸受到其上传送音频的媒介的峰值比特率、解码器的缓冲能力, 以及使每帧独立可解码的愿望的限制。

[0008] 通道内解相关(即去相关)12 通过对帧内每个通道中的音频采样解相关来去除冗余。大多数算法通过某种类型的信号线性预测建模来去除冗余。在这种方法中, 将线性预

测器应用于每帧中的音频采样，导致一系列预测误差采样。第二种不常用的方法是获得信号的低比特率量化或有损表示，并接着无损地压缩有损版本和原始版本之间的差异。熵编码 14 从来自残留信号的误差中去除冗余而不会丢失任何信息。典型的方法包括霍夫曼编码，行程长度编码和赖斯 (Rice) 编码。该输出是可能被无损地重建的压缩信号。

[0009] 现有的 DVD 规格和初级 HD DVD 规格对一个数据存取单元的尺寸设置了硬性限制，其代表一旦提取就能被完全解码的音频流的一部分，以及发送到输出缓冲器的重建的音频采样。对于无损流其含意是，每个存取单元可以代表的时间量必须足够小，以至于峰值比特率的最坏情况，编码净荷不超过该硬性限制。由于采样率增大和通道数量增加，这样就增大了峰比特率，持续时间必须缩短。

[0010] 为了保证兼容性，这些现有的编码器需要将整个帧的持续时间设置得足够短，以不超出最坏情况的通道 / 采样频率 / 比特宽度配置中的硬性限制。在大多数配置中，这将是不必要的过度行动并可能严重地劣化压缩性能。此外，这种最坏情况的方法不会随通道的增加而很好地变化 (scale)。

## 发明内容

[0011] 本发明提供了一种无损音频编解码器，其中在对每个独立可解码的数据单位的最大尺寸限制下优化压缩性能。

[0012] 无损音频编解码器将每帧内的音频数据分段，以在每个片段必须完全可解码并小于最大尺寸的约束条件下改善压缩性能。对于每帧而言，编解码器选择片段持续时间和编码参数，例如，用于每个片段的特定熵编码器和其参数，其在该约束下将整个帧的编码净荷降至最小。可以为每个通道选择不同的编码参数集，或者为所有通道选择全局性编码参数集。可以通过为 M- 通道音频形成  $M/2$  解相关通道来进一步增强压缩性能。通道的三重组（基准、相关、解相关）提供了可以在分段和熵编码优化期间考虑的两种可能的成对组合（基准、相关）和（基准、解相关）以进一步提高压缩性能。通道对可以是每段指定或每帧指定。

[0013] 在一个示例性的实施例中，编码器将音频数据成帧并随后提取包含基准通道和相关通道的有序通道对，并生成解相关通道以形成至少一个三重组（基准、相关、解相关）。如果通道的数量是奇数，则处理一个额外的基准通道。对每个通道应用自适应或固定多项式预测以形成残留信号。

[0014] 编码器通过首先将帧分割成具有最小持续时间的最大数量的片段，确定片段持续时间、该帧的通道对（（基准、相关）或（基准、解相关）），以及每个片段的编码参数集（熵代码选择和参数）。通过计算用于一个或多个熵编码器（二进制、赖斯、霍夫曼，等等）的参数，并为每个片段选择每个通道（基准、相关、解相关）具有最小编码净荷的编码器和参数来确定用于当前分割的最佳编码参数。对于每个三重组，选择具有最小编码净荷的通道对（基准、相关）或（基准、解相关）。利用所选择的通道对，可以为每个片段确定在全部通道上的全局性编码参数集。编码器基于哪个具有最小总编码净荷（头部和音频数据）来选择全局性编码参数集或不同的编码参数集。

[0015] 一旦已经确定了用于当前分割的编码参数集和通道对的最佳集，编码器计算每个片段中在所有通道上的编码净荷。假定满足对最大片段尺寸的约束条件，则编码器确定对

于当前分割的整个帧的总编码净荷是否小于对于较早分割的当前最佳。如果是，则存储编码参数和编码净荷的当前集并且增大片段持续时间。该过程重复直到片段尺寸打破了最大尺寸约束条件或者片段持续时间增长至帧持续时间。编码器（利用所选择的熵编码器和参数）对所选择的通道对和所有不成对的通道的每个音频通道中的残留信号熵编码。

[0016] 通过参照附图对以下优选实施例的详细描述，本发明的这些和其它特性和优点对于本领域的技术人员将是显而易见的。

## 附图说明

- [0017] 图 1, 如上所述, 是用于标准无损音频编码器的框图；
- [0018] 图 2a 和 2b 是根据本发明的无损音频编码器和解码器各自的框图；
- [0019] 图 3 是有关分段和熵代码选择的头部信息的示意图；
- [0020] 图 4a 和 4b 是分析窗口处理和逆向分析窗口处理的框图；
- [0021] 图 5 是交叉通道解相关的流程图；
- [0022] 图 6a 和 6b 是自适应预测分析和处理以及逆向自适应预测处理的框图；
- [0023] 图 7a 和 7b 是最佳分段和熵代码选择的流程图；
- [0024] 图 8a 和 8b 是用于通道集的熵代码选择的流程图；以及
- [0025] 图 9a 和 9b 是核心加无损扩展编解码器的框图。

## 具体实施方式

[0026] 本发明提供了一种无损音频编解码器，其中在对每个独立可解码的数据单位的最大尺寸约束条件下优化压缩性能。音频编码器随多通道音频中的通道数量的增加而进行调节。

### [0027] 无损音频编解码器

[0028] 如图 2a 和 2b 所示，除了分段和熵代码选择之外，基本的操作功能块类似于现有无损编码器和解码器。多通道 PCM 音频 20 经受分析窗口处理 22，其将数据分块为固定持续时间的帧并通过解相关帧内每个通道中的音频采样来去除冗余。代替直接对残留信号熵编码，本发明执行最佳分段和熵代码选择处理 24，其将数据分段成多个片段并为每个片段确定段持续时间和编码参数，例如，选择特定熵编码器和其参数，以便在每个片段必须完全可解码并小于最大尺寸的约束下使整个帧的编码净荷最小化。编码参数集被优化用于每个不同通道，也可以被优化用于全局性编码参数集。每个片段随后根据其不同的编码参数集被熵编码 26。编码数据和头部信息被打包 28 到比特流 30 中。

[0029] 如图 3 所示，头部 32 包除了正常为无损编解码器提供的以外的附加信息，以实现分段和熵代码选择。具体而言，该头部包括诸如片段数量 (NumSegments) 和每个片段中的采样数量 (NumSamplesInSegm) 的通用头部信息 34，诸如量化解相关系数 (QuantChDecorrrCoeff[][]) 的通道集头部信息 36 和诸如用于通道集的当前片段中的字节数量 (ChSetByteCOns)，指示是否使用了赖斯或二进制编码的全局优化标记 (AllChSameParamFlag) 和熵编码器标记 (RiceCodeFlag[], CodeParam[]) 以及编码参数的片段头部信息 38。

[0030] 如图 2b 所示，为了执行解码操作，比特流 30 被拆包 40 以提取头部信息和编码的

数据。根据分配的编码参数对每个通道的每个片段执行熵解码 42 以无损地重建残留信号。这些信号随后经受逆向分析窗口处理 44，其执行逆向预测以无损地重建原始 PCM 音频 20。

[0031] 分析窗口处理

[0032] 如图 4a 和 4b 所示，分析窗口处理 22 的示例性实施例或从自适应预测 46 选择或从固定多项式预测 48 选择以便解相关每个通道，这是一个相当普通的方法。如同将参照图 6 所详细描述的，为每个通道估计最佳预测器级数。如果该级数大于 0，则应用自适应预测。否则使用更简单的固定多项式预测。类似地，在解码器中，逆向分析窗口处理 44 要么从逆向自适应预测 50 选择，要么从逆向固定多项式预测 52 选择，以从残留信号重建 PCM 音频。该自适应预测器级数和自适应预测系数索引以及固定预测器级数被打包 53 到通道集头部信息中。

[0033] 交叉 - 通道解相关

[0034] 根据本发明，通过实现根据通道之间的相关测量将 M 个输入通道排序成通道对的交叉通道解相关 54，可以进一步增强压缩性能。其中一个通道被指定为“基准”通道，而另一个被指定为“相关”通道。为每个通道对生成解相关通道以形成“三重组”（基准、相关、解相关）。三重组的形成提供了两种可能的成对组合（基准、相关）和（基准、解相关），其可以在分段和熵编码优化期间被考虑以进一步提高压缩性能（参见图 8a）。一种更简单但是较低效率的方法将以解相关通道代替相关通道，例如，如果其方差较小的话。

[0035] 原始 M-ch PCM 20 和 M/2-ch 解相关 PCM 56 都被向前转发到为每个通道生成残留信号的自适应预测和固定多项式预测运算。如图 3 所示，在图 3 的通道集头部 36 中存储指示在成对解相关过程期间执行的排序之前的通道原始顺序的索引 (OrigChOrder[]) 以及指示用于量化解相关系数的代码的存在的、每个通道对的标记 PWChDecorrFlag[]。

[0036] 如图 4b 所示，为了执行逆向分析窗口处理 44 的解码操作，头部信息被拆包 58，并且根据头部信息，即每个通道的自适应和固定预测器级数，残留要么通过逆向固定多项式预测 52，要么通过逆向自适应预测 50。M- 通道解相关 PCM 音频 (M/2 通道在分段期间被丢弃) 通过逆向交叉通道解相关 60，其从通道集头部读取 OrigChOrder[] 索引和 PWChDecorrFlag[] 标记，并且无损地重建 M- 通道 PCM 音频 20。

[0037] 图 5 示意了用于执行交叉通道解相关 54 的一个实例性过程。举例来说，PCM 音频被提供为 M = 6 个不同通道：L、R、C、Ls、Rs 和 LFE，其也直接对应于存储在帧中的一个通道集配置。其它通道集例如可以是左中后环绕和右中后环绕，以产生 7.1 环绕音频。该过程是通过启动一个帧环路和启动一个通道集环路开始的（步骤 70）。计算对于每个通道的零滞后自相关估计（步骤 72）和对于通道集中通道对的所有可能组合的零滞后交叉相关估计（步骤 74）。其次，将通道成对相关系数 CORCOEF 估计为，零滞后交叉相关估计除以该通道对中包括的通道的零滞后自相关估计的乘积（步骤 76）。CORCOEF 按照从最大绝对值到最小绝对值排序并存储在表格中（步骤 78）。从表格的顶部开始，提取对应的通道对索引直到所有通道对都被配置（步骤 80）。例如，这 6 个通道可以基于它们的 CORCOEF 被配对为 (L, R), (Ls, Rs) 和 (C, LFE)。

[0038] 该过程启动一个通道对环路（步骤 82），并选择一个“基准”通道作为具有较小零滞后自相关估计的通道，较小零滞后自相关估计是较低能量的指示（步骤 84）。在这个实例中，L, Ls 和 C 通道形成基准通道。通道对解相关系数 (ChPairDecorrCoeff) 被计算为

零滞后交叉相关估计除以基准通道的零滞后自相关估计（步骤 86）。通过将基准通道采样与 CHPairDecorrrCoeff 相乘并从相关通道的对应采样减去该结果来生成解相关通道（步骤 88）。通道对和其相联系的解相关通道定义了“三重组”(L, R, R-ChPairDecorrrCoeff[1] \* L), (Ls, Rs, Rs-ChPairDecorrrCoeff[2] \* Ls), (C, LFE, LFE-ChPairDecorrrCoeff[3] \* C) (步骤 89)。用于每个通道对（以及每个通道集）的 ChPairDecorrrCoeff[] 和定义该对配置的通道索引被存储在通道集头部信息中（步骤 90）。该过程对于帧内的每个通道集重复，然后对窗口化 PCM 音频中的每帧重复（步骤 92）。

[0039] 自适应预测

[0040] 自适应预测分析和残留生成

[0041] 线性预测试图去除音频信号的采样之间的相关性。线性预测的基本原理是利用先前的采样  $s(n-1), s(n-2), \dots$  来预测采样值  $s(n)$  的值以及从原始采样  $s(n)$  中减去预测值  $\hat{s}(n)$ 。合成的残留信号  $e(n) = s(n) + \hat{s}(n)$  在理想情况下将是非相关的并因此具有平坦的频谱。另外，残留信号将具有更小的方差，于是该原始信号则暗示其数字表达式需要更少的比特。

[0042] 在音频编解码器的一个示例性实施例中，由以下公式描述 FIR 预测器模型：

$$[0043] e(n) = s(n) + Q \left\{ \sum_{k=1}^M a_k * s(n-k) \right\}$$

[0044] 其中， $Q \{\cdot\}$  表示量化操作， $M$  表示预测器级数，而  $a_k$  是量化的预测系数。由于原始信号是利用各种有限精度处理器体系结构在解码侧重建的，因此特殊量化  $Q \{\cdot\}$  是无损压缩所必需的。 $Q \{\cdot\}$  的定义对于编码器和解码器都是可用的，并且通过下式可以简单地获得原始信号的重建：

$$[0045] s(n) = e(n) - Q \left\{ \sum_{k=1}^M a_k * s(n-k) \right\}$$

[0046] 在此假定相同的  $a_k$  量化预测系数对于编码器和解码器都是可用的。每个分析窗口（帧）发送一组新的预测器参数，从而允许预测器适应于时间变化的音频信号结构。

[0047] 设计上述的预测系数以将均方预测残留降至最小。量化  $Q \{\cdot\}$  使预测器为非线性预测器。然而，在该示例性实施例中，该量化是以 24 位的精度完成的，而且假定在预测器系数优化期间可以忽略由此产生的非线性效应是合理的。通过忽略量化  $Q \{\cdot\}$ ，潜在的优化问题可以被表示为包括信号自相关序列的滞后和未知预测器系数的一组线性方程式。这组线性方程式可以利用 Levinson-Durbin (LD) 算法来有效地解出。

[0048] 由此产生的线性预测系数 (LPC) 需要被量化，以便它们可以有效地在编码流中发送。遗憾的是，由于小的量化误差可能引起大的频谱误差，LPC 的直接量化不是最有效的方法。LPC 的一个可选表达式是反射系数 (RC) 表达式，其对量化误差显现较小的灵敏度。还可以通过 LD 算法获得这种表达式。通过定义 LD 算法，保证 RC 具有量值  $\leq 1$  (忽略数值误差)。当 RC 的绝对值接近 1 时，对出现在量化的 RC 中的量化误差的线性预测的灵敏度变高。解决办法是以围绕单位 (unity) 的更精细量化级执行 RC 的非均匀量化。这可以通过两个步骤来完成：

[0049] 1) 将 RC 变换为 log-area 比 (LAR)

[0050] 借助映射函数表示

[0051]  $LAR = \log \frac{1+RC}{1-RC}$

[0052] 在此, log 表示自然底对数。

[0053] 2) 对 LAR 均匀量化

[0054]  $RC -> LAR$  变换歪曲了参数的幅度比例,使得步骤 1 和 2 的结果等效于具有围绕单位 (unity) 的更精细量化级的非均匀量化。

[0055] 如图 6a 所示,在自适应预测分析的示例性实施例中,量化的 LAR 参数用于表示自适应预测器参数并在编码比特流中发送。每个输入通道中的采样被相互独立地处理,因此该说明书将仅仅考虑在单个通道中的处理。

[0056] 第一步骤是计算分析窗口 (帧) 的持续时间内自相关序列 (步骤 100)。为了将由帧边界处的间断性引起的阻塞效应降至最小,数据首先被窗口化。从窗口化的数据块估计指定数量 (等于最大 LP 级数 +1) 的滞后的自相关序列。

[0057] Levinson-Durbin (LD) 算法被应用于该组估计的自相关滞后和该组反射系数 (RC),直到计算出最大 LP 级数 (步骤 102)。(LD) 算法的中间结果是对于高达最大 LP 级数的每个线性预测级数,预测残留的一组估计方差。在下一个功能块中,利用该组残留方差,选择线性预测器 (PrOr) 级数 (步骤 104)。

[0058] 对于所选择的预测器级数,利用以上陈述的映射函数将该组反射系数 (RC) 变换成该组 log-aria 比参数 (LAR) (步骤 106)。在变换之前引入 RC 的极限以防止除 0 :

[0059]

$$RC = \begin{cases} Tresh & \forall RC > Tresh \\ -1 & \forall RC < 1 \\ RC & \text{其他} \end{cases}$$

[0060] 在此, Tresh 表示接近但小于 1 的数。根据以下规则量化 LAR 参数 (步骤 108) :

[0061]

$$QLARInd = \begin{cases} \left\lfloor \frac{LAR}{q} \right\rfloor & \forall LAR \geq 0 \\ -\left\lfloor \frac{-LAR}{q} \right\rfloor & \forall LAR < 0 \end{cases}$$

[0062] 在此, QLARInd 表示量化的 LAR 索引,  $\lfloor x \rfloor$  指示找到小于或等于 x 的最大整数值的运算,而 q 表示量化级大小。在该示例性实施例中,利用 8 比特编码区域  $[-8 \text{ 至 } 8]$ ,即,

$q = \frac{2^*8}{2^8}$ ,因此 QLARInd 根据下式被限定 :

[0063]

$$QLARInd = \begin{cases} 127 & \forall QLARInd > 127 \\ -127 & \forall QLARInd < -127 \\ QLARInd & \text{其他} \end{cases}$$

[0064] 在打包之前 (步骤 110),利用以下映射将 QLARInd 从有符号值转换成无符号值 :

$$PackQLARInd = \begin{cases} 2^*QLARInd & \forall QLARInd \geq 0 \\ 2^*(-QLARInd)-1 & \forall QLARInd < 0 \end{cases}$$

[0066] 在“RC LUT”功能块中,在单个步骤中利用查找表完成 LAR 参数的逆向量化和对 RC 参数的转换(步骤 112)。查找表由逆向 RC->LAR 映射(即由下式给出的 LAR->RC 映射)的量化值组成:

$$[0067] \quad RC = \frac{e^{LAR} - 1}{e^{LAR} + 1}$$

[0068] 以等于  $0, 1.5 * q, 2.5 * q, \dots, 127.5 * q$  的 LAR 的量化值计算查找表。对应的 RC 值,在以  $2^{16}$  比例换算后,被四舍五入为 16 比特的无符号整数,并被存储作为 128 个条目的表中 Q16 个无符号的固定点数。

[0069] 量化的 RC 参数从该表中计算,且量化 LAR 索引 QLARInd 为:

$$[0070] \quad QRC = \begin{cases} TABLE[QLARInd] & \forall QLARInd \geq 0 \\ -TABLE[-QLARInd] & \forall QLARInd < 0 \end{cases}$$

[0071] 根据以下算法,量化的 RC 参数  $QRC_{ord}$  for  $ord = 1, \dots, PrOr$  被转换成量化的线性预测参数 ( $LP_{ord}$  for  $ord = 1, \dots, PrOr$ ) (步骤 114):

[0072] For  $ord = 0$  to  $PrOr-1$  do

[0073]     For  $m = 1$  to  $ord$  do

[0074]          $C_{ord+1, m} = C_{ord, m} + (QRC_{ord+1} * C_{ord, ord+1-m} + (1 << 15)) >> 16$

[0075]     end

[0076]      $C_{end+1, ord+1} = QRC_{ord+1}$

[0077] end

[0078] For  $ord = 0$  to  $PrOr-1$  do

[0079]      $LP_{ord+1} = C_{PrOr, ord+1}$

[0080] end

[0081] 由于量化的 RC 系数是以 Q16 有符号的固定点格式表示的,以上算法还将以 Q16 有符号的固定点格式生成 LP 系数。设计无损解码器计算路径以支持多达 24 比特的中间结果。因此需要在计算每个  $C_{ord+1, m}$  之后执行饱和检查。如果饱和发生在该算法的任何阶段,则设置饱和标记并且自适应预测器级数  $PrOr$ ,对于特殊通道,被复位为 0(步骤 116)。对于  $PrOr = 0$  的该特殊通道,将执行一个固定系数预测而不是自适应预测(参见固定系数预测)。注意,无符号的 LAR 量化索引 {PackLARIInd[n] for  $n = 1, \dots, PrOr[Ch]$ } 被打包成仅用于  $PrOr[ch] > 0$  的通道的编码流。

[0082] 最后对于  $PrOr > 0$  的每个通道,执行自适应线性预测,并且根据下式计算预测残留  $e(n)$ (步骤 118):

$$[0083] \quad \overline{s(n)} = \left[ \left\{ \sum_{k=1}^{PrOr} LP_k * s(n-k) \right\} + (1 << 15) \right] >> 16$$

[0084] Limit  $\overline{s(n)}$  to 24-bit range ( $-2^{23}$  to  $2^{23}-1$ )

$$[0085] \quad e(n) = s(n) + \overline{s(n)}$$

[0086] Limit  $e(n)$  to 24-bit range ( $-2^{23}$  to  $2^{23}-1$ )

[0087] for  $n = PrOr+1, \dots, NumSampUnFrame$

[0088] 由于该示例性实施例中的设计目标是每个帧是“随机存取点”,采样历史不会在帧

之间延续。取而代之的是，仅在该帧中的 PrOr+1 采样处进行预测。

[0089] 自适应预测残留 e(n) 被进一步熵编码并被打包成编码的比特流。

[0090] 解码侧的逆向自适应预测

[0091] 在解码侧，执行逆向自适应预测的第一步是将头部信息拆包并提取用于每个通道 Ch = 1, ..., NumCh 的自适应预测级数 PrOr[Ch]（步骤 120）。接下来，对于 PrOr[Ch] > 0 的通道，提取 LAR 量化索引 (PackLARIInd[n] for n = 1, ..., PrOr[Ch]) 的无符号版本。对于具有预测级数 PrOr[Ch] > 0 的每个通道 Ch，利用以下映射将无符号的 PackLARIInd[n] 映射为有符号的值 QLARIInd[n]：

$$[0092] QLARIInd[n] = \begin{cases} PackLARIInd[n] \gg 1 & \forall \text{even numbered } PackLARIInd[n] \\ -(PackLARIInd[n] \gg 1) - 1 & \forall \text{odd numbered } PackLARIInd[n] \end{cases}$$

[0093] for n = 1, ..., PrOr[Ch]

[0094] 在此，>> 表示整数右移运算。

[0095] 在单个步骤中利用 Quant RC LUT 完成 LAR 参数的逆向量化和对 RC 参数的转换（步骤 122）。它是与在编码侧定义查找表的相同的查找表 TABLE{}。通过 TABLE{} 和量化 LAR 索引 QLARIInd[n] 计算对于每个通道 Ch (QRC[n] for n = 1, ..., PrOr[Ch]) 的量化反射系数：

$$[0096] QRC[n] = \begin{cases} TABLE[QLARIInd[n]] & \forall QLARIInd[n] \geq 0 \\ -TABLE[-QLARIInd[n]] & \forall QLARIInd[n] < 0 \end{cases}$$

[0097] for n = 1, ..., PrOr[Ch]

[0098] 对于每个通道 Ch，根据以下算法，量化的 RC 参数 QRC<sub>ord</sub> for ord = 1, ..., PrOr[Ch] 被转换成量化的线性预测参数 (LP<sub>ord</sub> for ord = 1, ..., PrOr[Ch])（步骤 124）：

[0099] For ord = 0 to PrOr-1 do

[0100] Form = 1 to ord do

[0101] C<sub>ord+1, m</sub> = C<sub>ord, m</sub> + (QRC<sub>ord+1</sub> \* C<sub>ord, ord+1-m</sub> + (1 << 15)) >> 16

[0102] end

[0103] C<sub>ord+1, ord+1</sub> = QRC<sub>ord+1</sub>

[0104] end

[0105] For ord = 0 to PrOr-1 do

[0106] LP<sub>ord+1</sub> = C<sub>PrOr, ord+1</sub>

[0107] end

[0108] 中间结果的任何饱和可能性在编码侧被去除。因此在解码侧，在计算每个 C<sub>ord+1, m</sub> 之后不需要执行饱和检查。

[0109] 最后对于 PrOr[Ch] > 0 的每个通道，执行逆向自适应线性预测（步骤 126）。假设预测残留 e(n) 被预先提取并被熵解码，根据下式计算重建原始信号 s(n)：

$$[0110] \overline{s(n)} = \left[ \left\{ \sum_{k=1}^{PrOr[Ch]} LP_k * s(n-k) \right\} + (1 << 15) \right] >> 16$$

[0111] Limit  $\overline{s(n)}$  to 24-bit range (-2<sup>23</sup> to 2<sup>23</sup>-1)

$$[0112] e(n) = s(n) - \overline{s(n)}$$

[0113] for  $n = \text{PrOr}[\text{Ch}]+1, \dots, \text{NumSampUnFrame}$

[0114] 由于采样历史没有在帧之间被保持, 逆向自适应预测将从该帧中的 ( $\text{PrOr}[\text{Ch}]+1$ ) 采样开始。

[0115] 固定系数预测

[0116] 已经发现线性预测器的一种非常简单的固定系数格式会非常有用。该固定预测系数是根据由 Shorten 首先提出的非常简单的多项式近似方法得出的 (T. Robinson. SHORTEN :Simple lossless and near lossless waveform compression. Technical report 156. Cambridge University Engineering Department Trumpington Street, Cambridge CB21PZ, UK December 1994)。在这种情况下, 预测系数是那些通过将一个 P 阶多项式提供给最后 P 个数据点指定的预测系数。在以下四个近似式上展开 :

$$\hat{s}_0[n] = 0$$

$$\hat{s}_1[n] = s[n-1]$$

$$\hat{s}_2[n] = 2s[n-1] - s[n-2]$$

$$\hat{s}_3[n] = 3s[n-1] - 3s[n-2] + s[n-3]$$

[0121] 这些多项式近似的有趣特性是, 最终产生的残留信号  $e_k[n] = s[n] - \hat{s}_k[n]$  可以以下列递归方式有效地实现。

$$e_0[n] = s[n]$$

$$e_1[n] = e_0[n] - e_0[n-1]$$

$$e_2[n] = e_1[n] - e_1[n-1]$$

$$e_3[n] = e_2[n] - e_2[n-1]$$

[0126] 固定系数预测分析以每帧为基础应用, 并且不依赖在先前帧 ( $e_k[-1] = 0$ ) 中计算的采样。在整个帧内具有最小和数量值的残余集被定义为最佳近似值。为每个通道分别计算最佳残差级并将其打包成流作为固定预测级 (FPO[Ch])。当前帧内的残留  $e_{\text{FPO}[\text{Ch}]}$  [n] 被进一步熵编码并打包成流。

[0127] 反向固定系数预测处理, 在解码侧, 通过顺序递归公式定义, 用于在采样时刻 n 计算 k 阶残留 :

$$e_k[n] = e_{k+1}[n] + e_k[n-1]$$

[0129] 在此, 期望的原始信号  $s[n]$  通过下式给出 :

$$s[n] = e_0[n]$$

[0131] 并且在此对于每个 k 阶残留,  $e_k[-1] = 0$ 。

[0132] 举例来说, 提供用于第 3 级固定系数预测的递归, 在此残留  $e_3[n]$  被编码, 以流发送并在解码侧被拆包 :

$$e_2[n] = e_3[n] + e_2[n-1]$$

$$e_1[n] = e_2[n] + e_1[n-1]$$

$$e_0[n] = e_1[n] + e_0[n-1]$$

$$s[n] = e_0[n]$$

[0137] 分段和熵代码选择

[0138] 图 7 和 8 示意了分段和熵代码选择 24 的一个示例性实施例。为了设定最佳片段

持续时间、编码参数（熵代码选择 & 参数）和通道对，为多个不同的片段持续时间确定编码参数和通道对，并且在那些候选对象中，选择每帧具有最小编码净荷、且满足每个片段必须独立可解码并不超过最大尺寸的约束条件的一个候选对象。“最佳”分段、编码参数和通道对自然受到编码处理的约束以及对片段尺寸的约束。例如，在该示例性的处理中，帧内所有片段的持续时间相等，在并矢格栅上执行对最佳持续时间的检索，并且通道对选择在整个帧上都是有效的。以额外的编码器复杂性和开销比特为代价，可以允许持续时间在帧内变化，可以更好地解决对最佳持续时间的检索，并且可以基于每片段完成通道对选择。

[0139] 该示例性的处理是以初始化诸如片段内的采样的最小数量、片段的最大允许尺寸、片段的最大数量和分割的最大数量的片段参数开始的（步骤 150）。此后，该处理启动一个分割环路，其索引是从 0 到分割的最大数量减 1（步骤 152），并且初始化包含片段数量、片段内的采样数量和在分割内消耗的字节数量的分割参数（步骤 154）。在该特殊实施例中，这些片段具有相等的持续时间，并且片段的数量随着每次分割迭代换算为二的幂，片段的数量优选被初始化为最大值，因此初始化为最小的持续时间。然而，该处理可以使用持续时间变化的片段，其可能提供更好的音频数据压缩，但是以额外的开销为代价。此外，片段的数量不限于二的幂或者从最小到最大持续时间进行检索。

[0140] 一旦初始化，该处理启动一个通道集环路（步骤 156），并为每个片段以及对应的字节消耗确定最佳熵编码参数和通道对选择（步骤 158）。编码参数 PWChDecorFlag[][], A11ChSameParamFlag[][], RiceCodeFlag[][], CodeParam[][][] 和 ChSetByteCons[][][] 被存储（步骤 160）。对于每个通道集重复该处理，直到通道集环路结束（步骤 162）。

[0141] 该处理启动一个片段环路（步骤 164），并计算在所有通道集上每个片段中的字节消耗 (SegmByteCons)（步骤 166）并更新字节消耗 (ByteConsInPart)（步骤 168）。在该点处，将片段的尺寸与最大尺寸约束相比较（步骤 170）。如果该约束被打破，则丢弃当前分割。此外，由于该处理是以最小持续时间开始的，一旦片段尺寸太大，则分割环路终结（步骤 172）且对于该点的最佳解决办法（持续时间、通道对、编码参数）被打包到头部内（步骤 174）并且该处理转至下一帧。如果对最小片段上尺寸的约束失败（步骤 176），则由于不能满足最大尺寸约束，该处理终结并报告一个错误（步骤 178）。假定满足该约束，则对于当前分割内的每个片段重复该处理直到片段环路结束（步骤 180）。

[0142] 一旦片段环路已经完成，且整个帧的字节消耗被计算为由 ByteConsInPart 表示，该净荷与来自先前分割迭代的当前最小净荷 (MinByteInPart) 相比较（步骤 182）。如果当前分割表示一个改进，那么当前分割 (PartInd) 被存储作为最佳分割 (OptPartInd) 并且更新最小净荷（步骤 184）。这些参数和所存储的编码参数随后被存储作为当前最佳解决方案（步骤 186）。这将重复一直到分割环路结束（步骤 172），如图 3 所示，在该点分段信息和编码参数被打包到头部中（步骤 150）。

[0143] 图 8a 和 8b 示意了用于为当前分割的通道集确定最佳编码参数和相关比特消耗的一个示例性实施例（步骤 158）。该处理启动一个分段环路（步骤 190）和通道环路（步骤 192），其中用于我们当前实例的通道是：

[0144] Ch1 :L,

[0145] Ch2 :R

[0146] Ch3 :R-ChPairDecorCoeff[1] \* L

[0147] Ch4 :Ls

[0148] Ch5 :Rs

[0149] Ch6 :Rs-ChPairDecorrCoeff[2] \* Ls

[0150] Ch7 :C

[0151] Ch8 :LFE

[0152] Ch9 :LFE-ChPairDecorrCoeff[3] \* C)

[0153] 该处理为基准和相关通道确定熵代码的类型,对应的编码参数和对应的比特消耗(步骤 194)。在这个实例中,该处理计算用于二进制代码和赖斯代码的最佳编码参数,然后为通道和每个片段选择具有最低比特消耗的一个(步骤 196)。通常,可以对一个、两个或多个可能的熵代码进行优化。对于二进制代码,从当前通道的片段中的所有采样的最大绝对值计算比特数量。从当前通道的片段中的所有采样的平均绝对值计算赖斯编码参数。基于该选择,设置 RiceCodeFlag, 设置 BitCons, 并设置 CodeParam 为 NumBitsBinary 或 RiceKParam(步骤 198)。

[0154] 如果被处理的当前通道是相关通道(步骤 200),则对对应的解相关通道重复相同的优化(步骤 202),选择最佳熵代码(步骤 204)并且设置编码参数(步骤 206)。该处理重复直到通道环路结束(步骤 208)和片段环路结束(步骤 210)。

[0155] 在该点处,已经确定用于每个片段和用于每个通道的最佳编码参数。可以为通道对(基准、相关)从原始 PCM 音频返回这些编码参数和净荷。然而,可以通过在三重组中的(基准、相关)和(基准、解相关)通道之间进行选择来提高压缩性能。

[0156] 为了确定哪个通道对(基准、相关)或(基准、解相关)用于三个三重组,启动一个通道对环路(步骤 211),并计算每个相关通道(Ch2、Ch5 和 Ch8)和每个解相关通道(Ch3、Ch6 和 Ch9)对总帧比特消耗的贡献(步骤 212)。将对每个相关通道作贡献的帧消耗贡献与对对应的解相关通道作贡献的帧消耗贡献相比较,即,Ch2 对 Ch3, Ch5 对 Ch6, 以及 Ch8 对 Ch9(步骤 214)。如果解相关通道的贡献大于相关通道,则设置 PWChDecorrFlag 为假(步骤 216)。否则,以解相关通道代替相关通道(步骤 218)并且设置 PWChDecorrFlag 为真,而且通道对被配置为(基准、解相关)(步骤 220)。

[0157] 基于这些比较,该算法将选择:

[0158] 1. Ch2 或 Ch3 作为将与对应的基准通道 Ch1 成对的通道;

[0159] 2. Ch5 或 Ch6 作为将与对应的基准通道 Ch4 成对的通道;以及

[0160] 3. Ch8 或 Ch9 作为将与对应的基准通道 Ch7 成对的通道。

[0161] 这些步骤对所有的通道对重复直到该环路结束(步骤 222)。

[0162] 在该点处,已经确定了用于每个片段和每个不同通道以及最佳通道对的最佳编码参数。用于每个不同的通道对和净荷的这些编码参数可以被返回到分割环路。然而,可以通过为每个片段计算在所有通道上一个全局性编码参数集来得到附加的压缩性能。最好,净荷的编码数据部分将与对每个通道优化的编码参数具有相同的尺寸,并且很有可能稍大。然而,开销比特的减少可能大于对数据编码效率的抵消。

[0163] 利用相同的通道对,该处理启动一个片段环路(步骤 230),利用不同的编码参数集针对所有通道计算每个片段的比特消耗(ChSetByteCons[seg])(步骤 232)并存储 ChSetByteCons[seg](步骤 234)。随后利用与之前的非用于所有通道的相同的

二进制代码和赖斯代码计算,为片段确定在所有通道上的全局性编码参数(熵代码选择和参数)集(步骤 236)。选择最佳参数,并且计算字节消耗(SegmByteCons)(步骤 238)。将 SegmByteCons 与 CHSetByteCons[seg] 相比较(步骤 240)。如果利用全局参数没有减少比特消耗,则将 AllChSamParamFlag[seg] 设置为假(步骤 242)。否则,将 AllChSameParamFlag[seg] 设置为真(步骤 244),并且保存全局编码参数和对应的每片段比特消耗(步骤 246)。该过程重复直至片段环路的结束(步骤 248)。整个处理重复直到通道集环路终结步骤 250。

[0164] 编码处理是以可以通过少数标记的控制来禁用不同的功能的方式来构建的。例如,一个单个标记控制是否执行成对通道解相关分析。另一个标记控制是否执行自适应预测(还有另一个标记用于固定预测)分析。另外,单个标记控制是否执行所有通道上的全局参数的检索。通过设置分割的数量和最小片段持续时间(在最简单的形式下,可以是具有预定片段持续时间的单个分割),分段也是可以控制的。本质上,通过在编码器中设置少量标记,该编码器可以破坏简单成帧和熵编码。

[0165] 后向兼容无损音频编解码器

[0166] 无损编解码器可以用作与有损核心编码器结合的“扩展编码器”。“有损”核心代码流被打包为核心比特流,而无损编码的差分信号被打包为分离的扩展比特流。一旦在具有扩展的无损特性的解码器中解码,有损和无损流被合并以构建无损重建信号。在前一代解码器中,无损流被忽略,并且核心“有损”流被解码以提供具有核心流的带宽和信噪比特性的高质量、多通道音频信号。

[0167] 图 9 示出了用于多通道信号的一个通道的后向兼容无损编码器 400 的系统级视图。在输入端 402 提供数字化的音频信号,适当的为 M 比特的 PCM 音频采样。优选地,数字化的音频信号具有超出修正的有损核心编码器 404 的采样率和带宽。在一个实施例中,数字化音频信号的采样率是 96kHz(对应于采样音频的 48kHz 的带宽)。还应当理解的是,输入音频可以以及优选为多通道信号,其中每个通道以 96kHz 采样。下面将集中讨论单个通道的处理,但是扩展到多通道是直接了当的。输入信号在节点 406 被复制,并且在并行分支中被处理。在信号路径的第一个分支中,一个修正的有损、宽带编码器 404 编码该信号。在下面详细描述的修正的核心编码器 404 产生编码核心比特流 408,其被传送到打包器或复用器 410。核心比特流 408 还被传送到修正的核心解码器 412,该核心解码器产生修正的重建核心信号 414 作为输出。

[0168] 同时,并行路径中的输入数字化音频信号 402 经历补偿延迟 416,该补偿延迟基本上等于(通过修正的编码器和修正的解码器)引入到重建音频流中的延迟,以产生延迟的数字化音频流。在加法节点 420 从延迟的数字化音频流 414 减去音频流 400。求和节点 420 产生表示原始信号和重建核心信号的差分信号 422。为了完成纯“无损”编码,需要以无损编码技术编码和发送该差分信号。因此,用无损编码器 424 编码差分信号 422,并在打包器 410 中将扩展比特流 426 与核心比特流 408 打包,以产生输出比特流 428。

[0169] 注意,无损编码产生可变比特率的扩展比特流 426,以适应无损编码器的需要。被打包的流随后任选地经过包括通道编码的编码的其它层,并且随后被发送或记录。注意,为了本公开内容的目的,记录可以被认为是通过通道传输的。

[0170] 核心编码器 404 被描述为“修正的”,这是因为在能够处理扩展带宽的实施例中,

核心编码器将需要修正。编码器内的 64- 频带分析滤波器组 430 丢弃其输出数据 432 的一半，并且核心子带编码器 434 仅仅编码较低的 32 个频带。这个丢弃的信息对于在任何情况下都不能重建信号谱上半部分的传统解码器而言是不涉及的。剩余信息经由未修正的编码器编码以形成后向兼容的核心输出流。然而，在以 48kHz 或低于 48kHz 的采样率工作的另一个实施例中，核心编码器可以是现有核心编码器的基本上未修正的版本。类似地，为了在上述的传统解码器的采样率以上工作，修正的核心解码器 412 包括以较低的 32 个子带解码采样的核心子带解码器 436。修正的核心解码器从较低的 32 个子带取出子带采样，并且对于较高的 32 个频带 438 使未发送的子带采样归零，并且利用 64- 频带 QMF 合成滤波器 440 重建所有的 64 个频带。为了以常规采样率（例如，48kHz 和 48kHz 以下）操作，核心解码器可以是现有核心解码器的基本上未修正版本或等效物。在一些实施例中，可以在编码时选择采样率，并且如同期望的那样，在那时通过软件重新配置编码和解码模块。

[0171] 由于无损编码器将被用于编码差分信号，似乎简单的熵编码就足够了。然而，由于对现有有损核心编解码器的比特率限制，仍然需要大量的总比特以提供无损比特流。此外，由于核心编解码器的带宽限制，该差分信号中的 24kHz 以上的信息内容仍然相关。例如，大量的谐波成分，包括喇叭、吉他、三角铁...，远远超过 30kHz。因此更为成熟的提高了压缩性能的无损编解码器可增值。另外，在一些应用中，核心和扩展比特流必须仍然满足可解码单位必须不超出最大尺寸的约束。本发明的无损编解码器同时提供了提高的压缩性能和改善的灵活性以满足这些约束。

[0172] 举例来说，8 个通道的 24 比特 96kHz PCM 音频需要 18.5Mbps。无损压缩可以将其降低到大约 9Mbps。DTS 相关声学器件将以 1.5Mbps 编码核心，剩下 7.5Mbps 的差分信号。对于 2K 字节的最大片段尺寸，平均片段持续时间为  $2048 * 8 / 7500000 = 2.18\text{msec}$  或者大约 209 个采样 @96kHz。满足该最大尺寸的有损核心的典型帧尺寸在 10 到 20msec 之间。

[0173] 可以在系统级合并无损编解码器和后向兼容无损编解码器以在扩展带宽处无损地编码额外的音频通道，同时保持与现有有损编解码器的后向兼容性。例如，以 18.5Mbps 的 8 个通道的 96kHz 音频可以被无损地编码以包括以 1.5Mbps 的 5.1 通道的 48kHz 音频。核心加无损编码器将用于编码 5.1 通道。无损编解码器将用于编码 5.1 通道中的差分信号。剩余的 2 个通道利用无损编码器以分开的通道集编码。由于当试图优化片段持续时间时需要考虑所有通道集，所有的编码工具将以一种方式或另一种方式使用。一个兼容的解码器将解码所有 8 个通道并无损地重建 96kHz 的 18.5Mbps 音频信号。而较老的解码器将仅仅解码 5.1 通道并重建 48kHz 1.5Mbps。

[0174] 一般而言，为了计算解码器的复杂性，可以提供一个以上纯无损通道集。例如，对于 10.2 原始混频，可以如此来组织通道集：

[0175] -CHSET1 携带 5.1（具有嵌入的 10.2 到 5.1 下混频）并且利用核心 + 无损编码

[0176] -CHSET1 和 CHSET2 携带 7.1（具有嵌入的 10.2 到 7.1 下混频），在此，CHSET2 利用无损编码 2 个通道

[0177] -CHSET1+CHSET2+CHSET3 携带全离散 10.2 混频，在此，CHSET3 仅利用无损编码剩余的 3.1 通道

[0178] 能够仅解码 5.1 的解码器将仅解码 CHSET1 并忽略所有其它通道集。能够仅解码 7.1 的解码器将解码 CHSET1 和 CHSET2 并忽略所有其它通道集。...

[0179] 此外,有损加无损核心不局限于 5.1。当前实现方案利用有损 (core+XCh) 和无损支持高达 6.1,并且可以支持以任意数量的通道集组织的普通 m. n 通道。有损编码将具有 5.1 后向兼容核心,并且用有损编解码器编码的所有其它通道将进入 XXCh 扩展。这就提供了具有相当大的设计灵活性的总体无损编码,以保持与现有解码器的后向兼容,同时支持附加通道。

[0180] 虽然已经示出并描述了本发明的多个示例性实施例,本领域的技术人员将想到多种变型和可选实施例。这些变型和可选实施例是可预期的,并且可以不偏离所附权利要求书中定义的本发明的精神和范围做出这些变型和可选实施例。

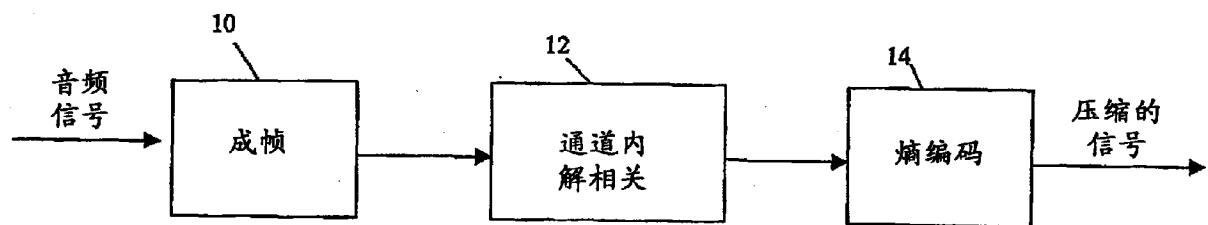


图1(现有技术 )

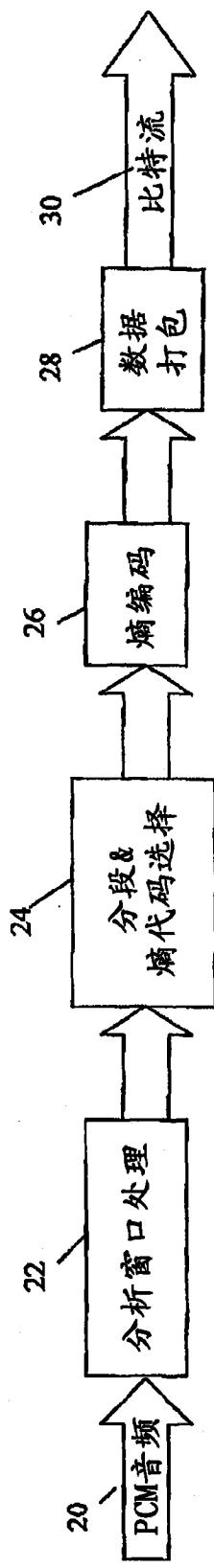


图 2a

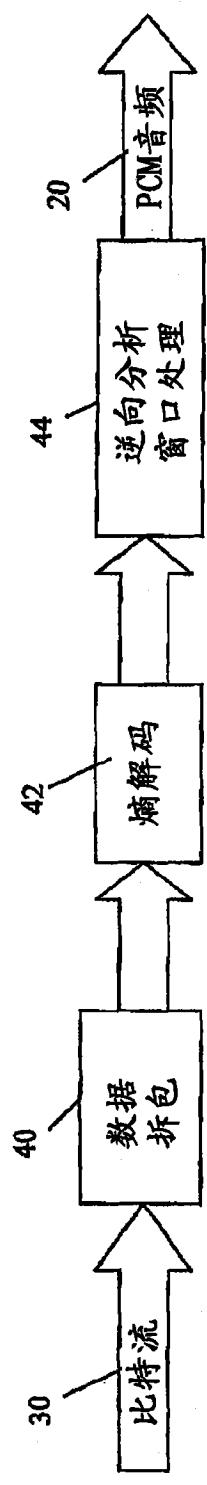
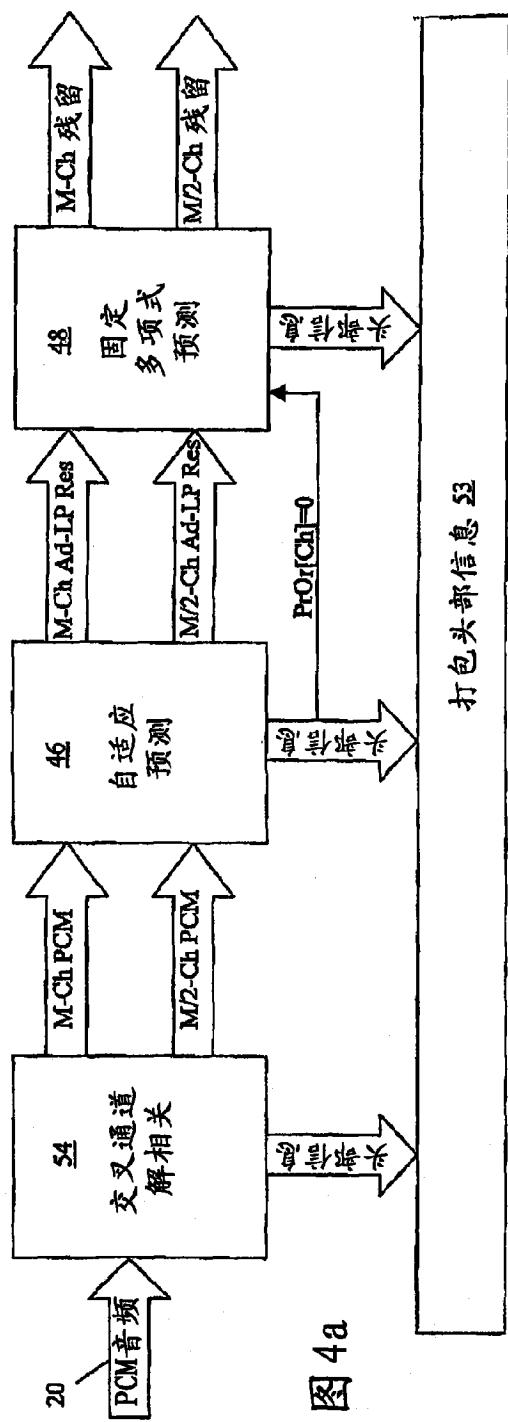
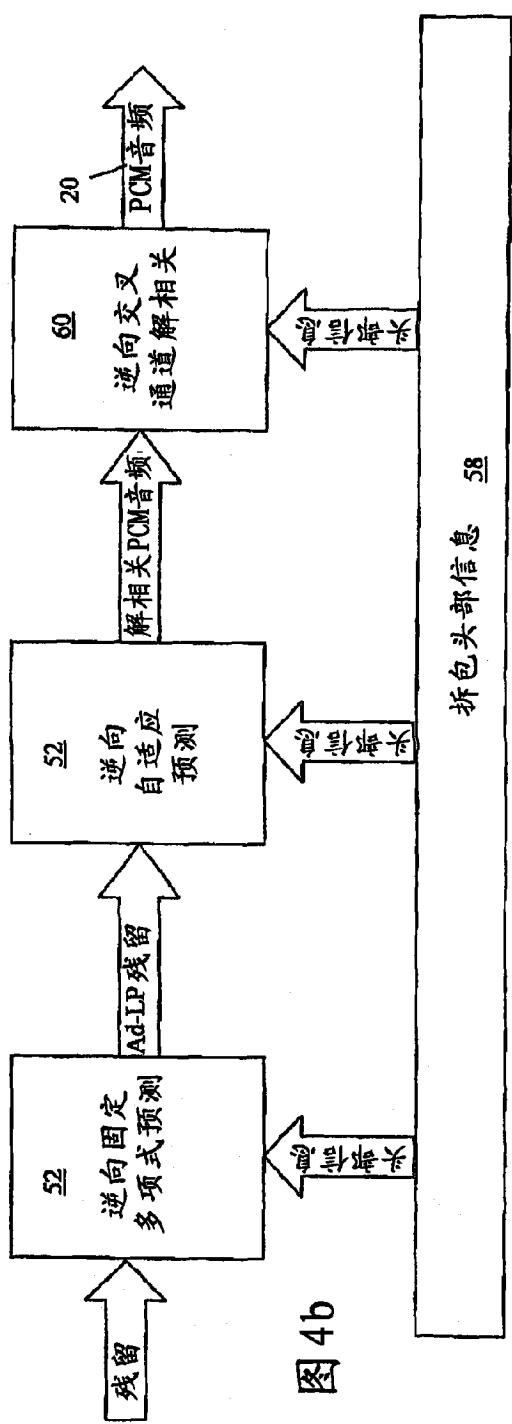


图 2b



打包头部信息 53



拆包头部信息 58

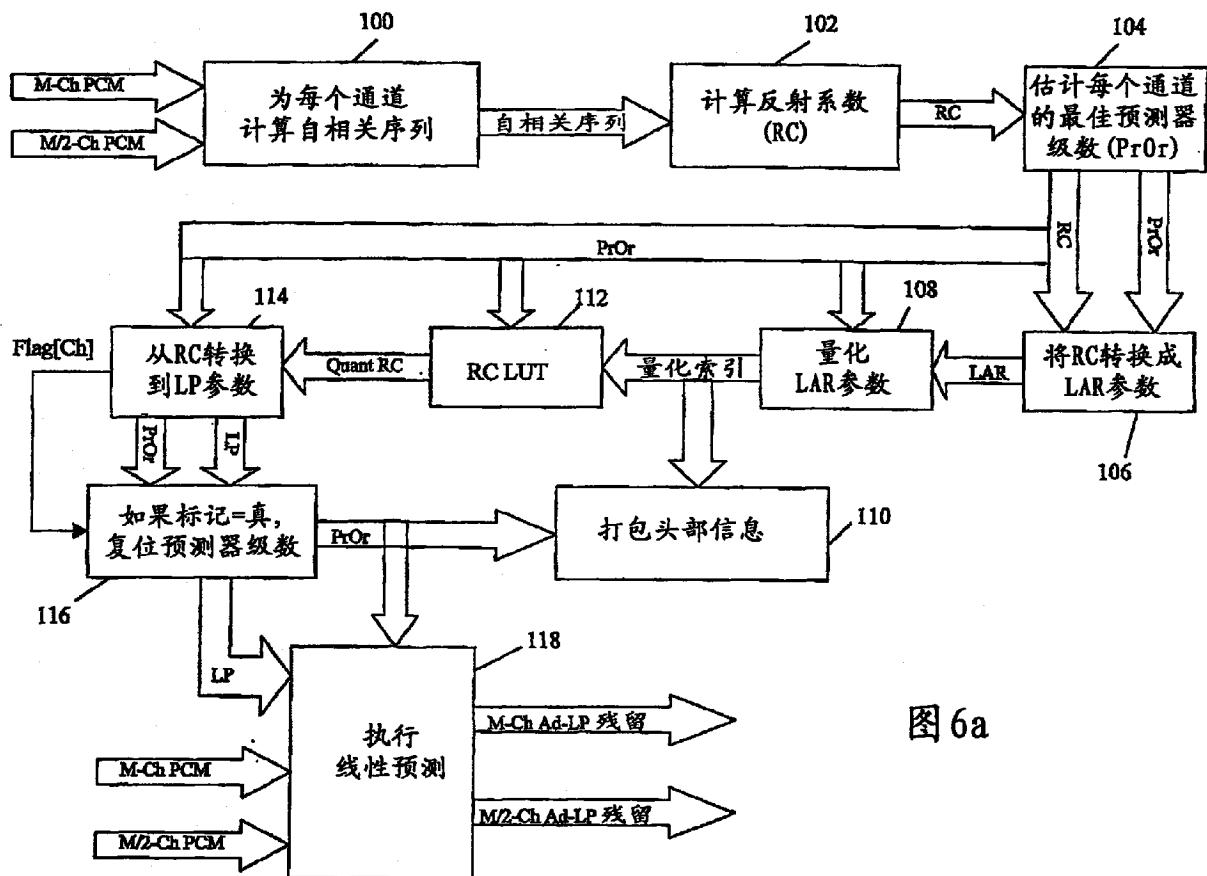


图 6a

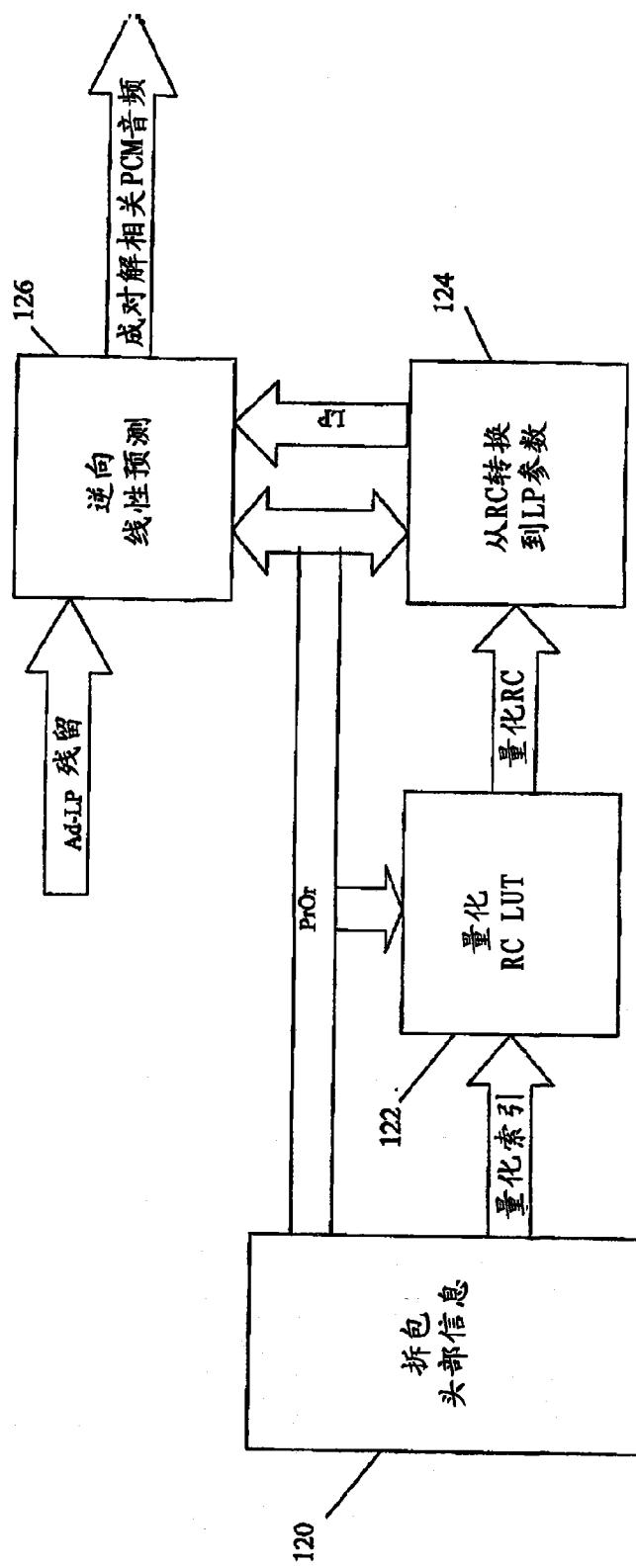


图 6b

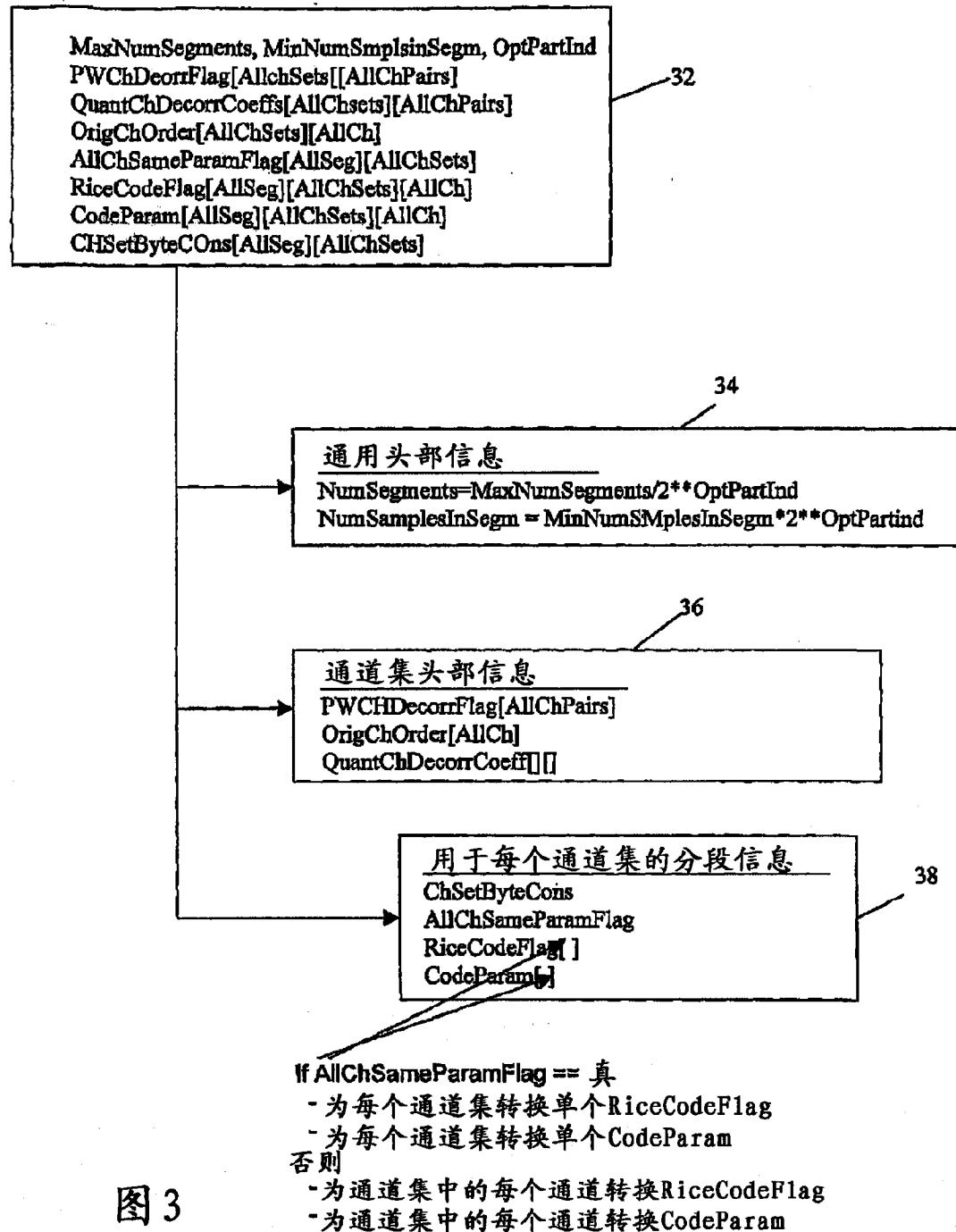
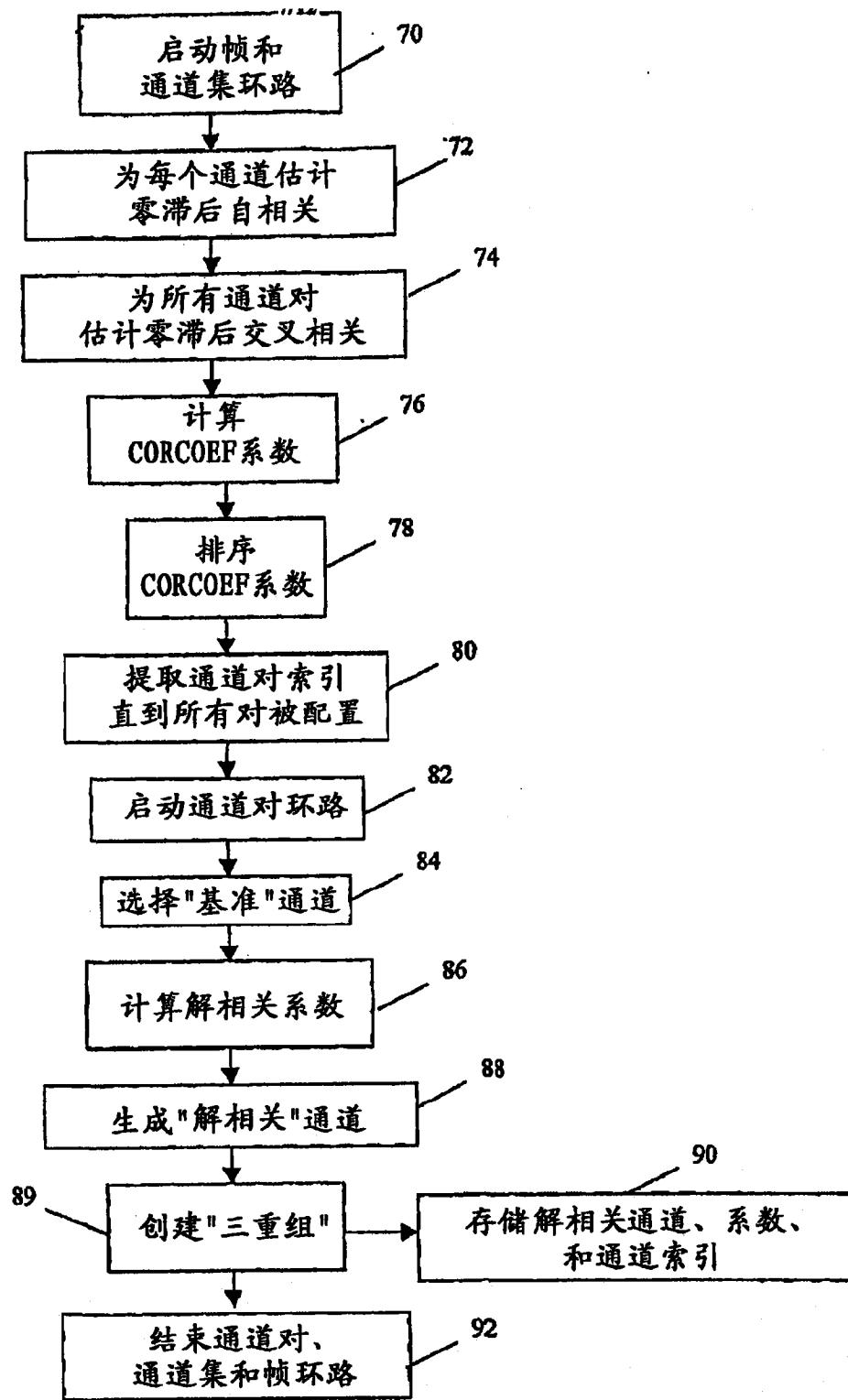
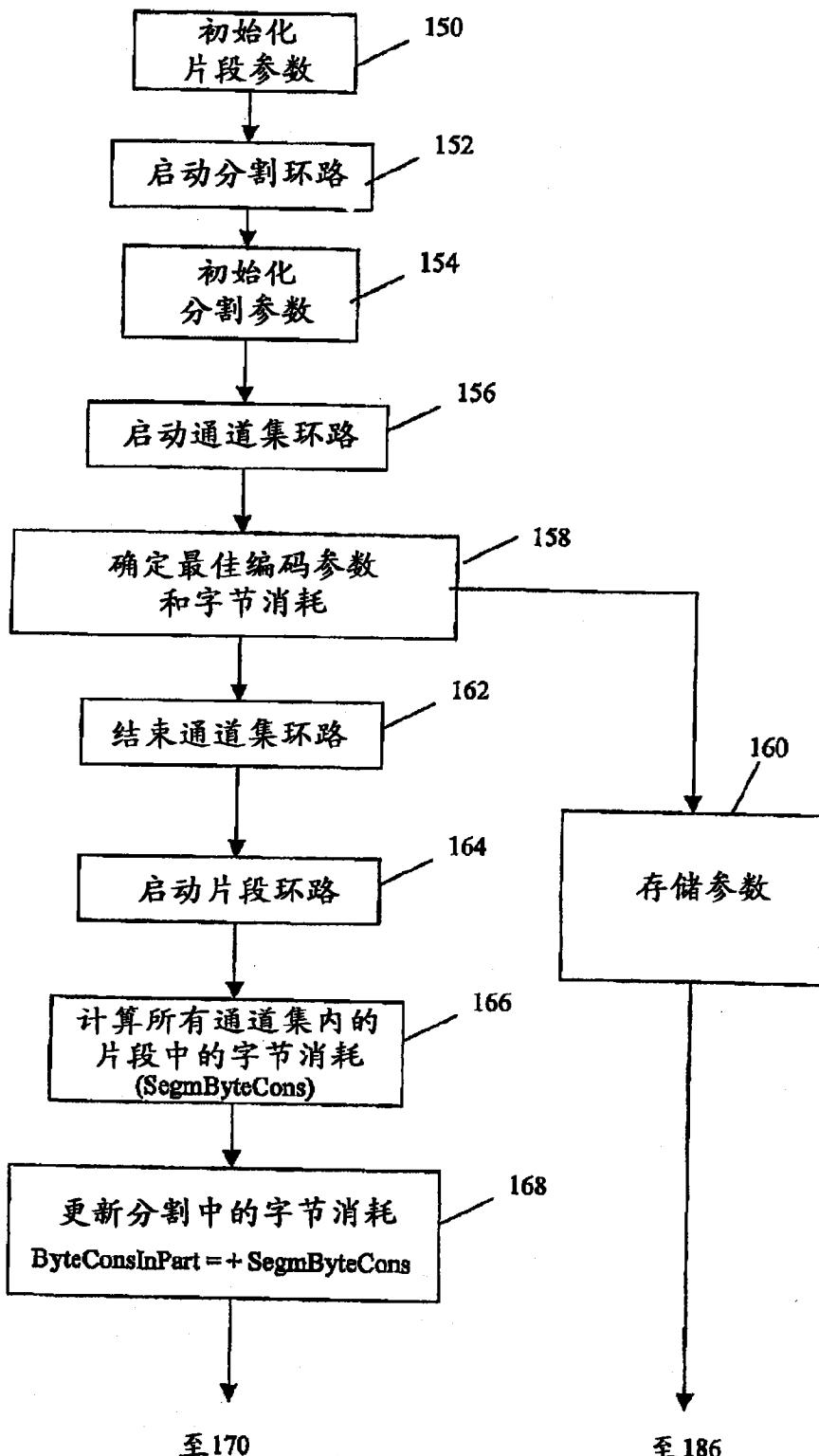


图 3





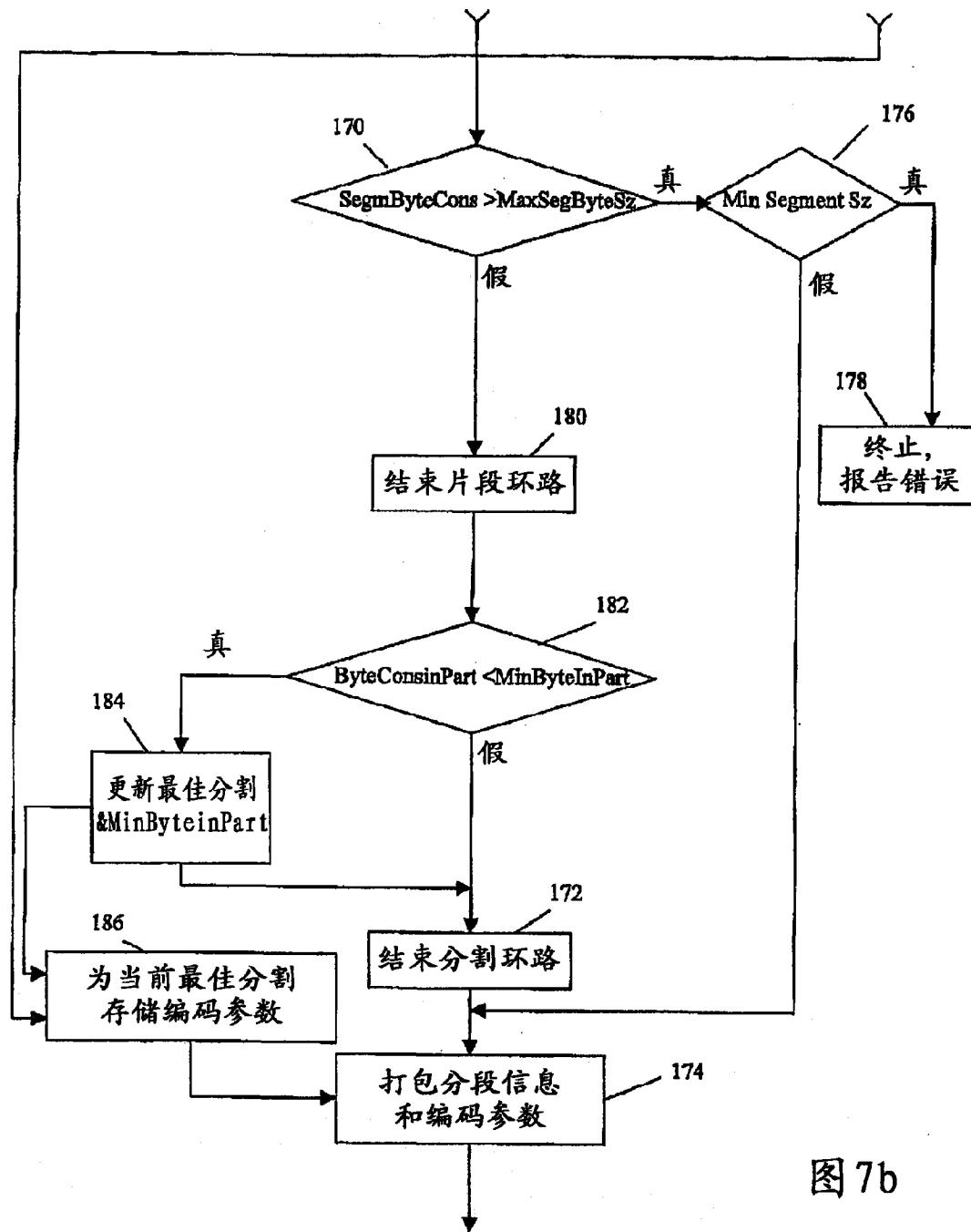
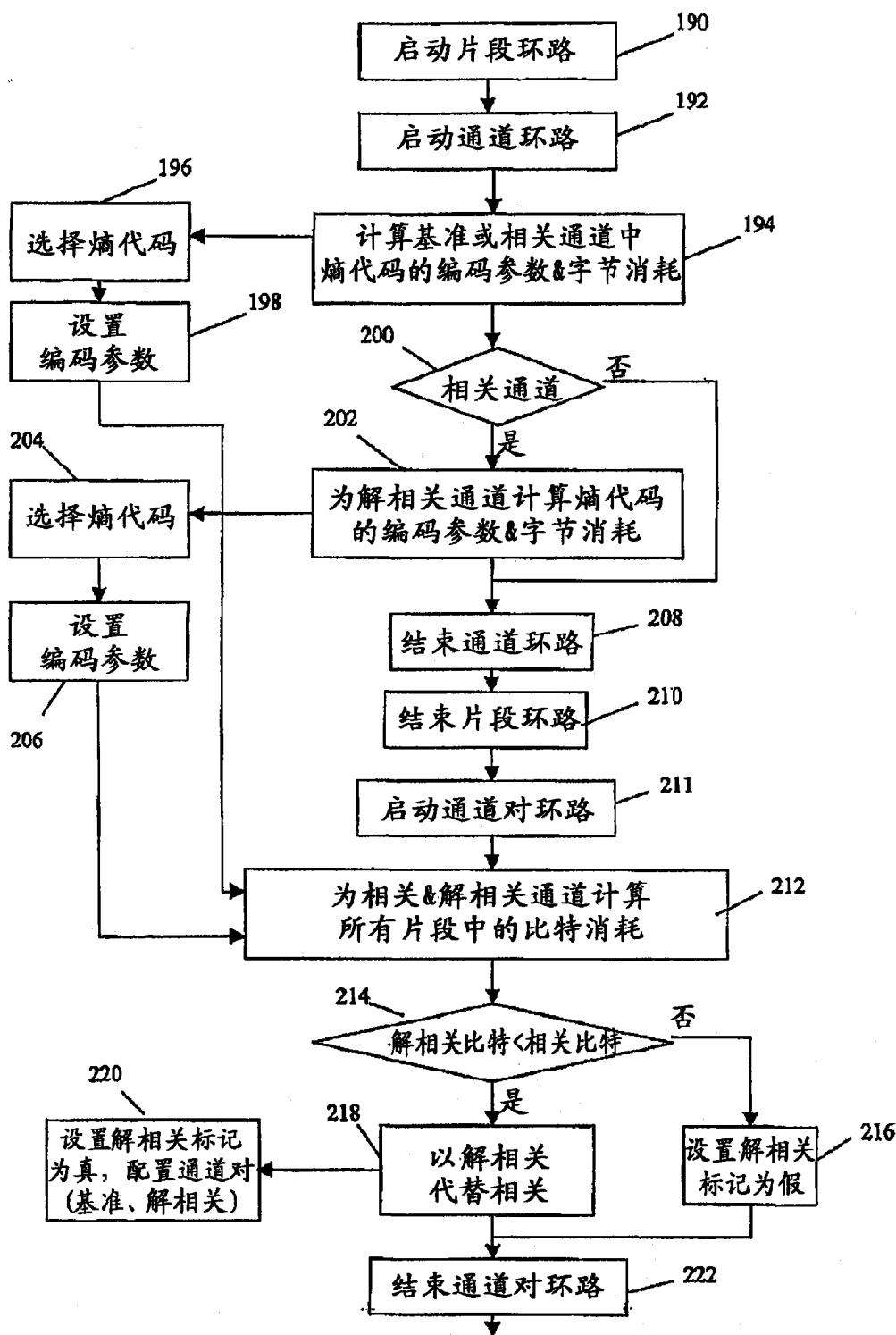
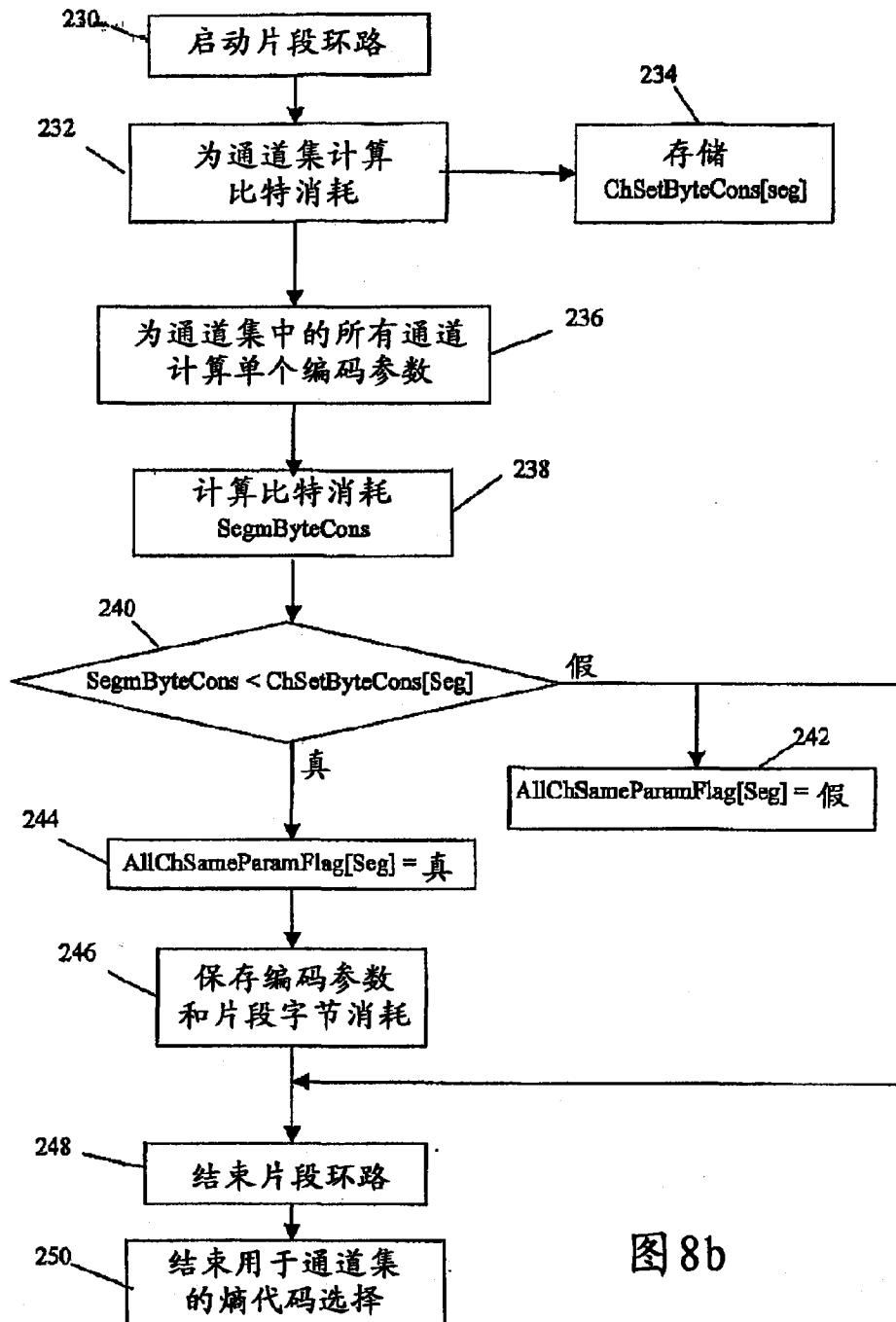


图 7b

图 8a





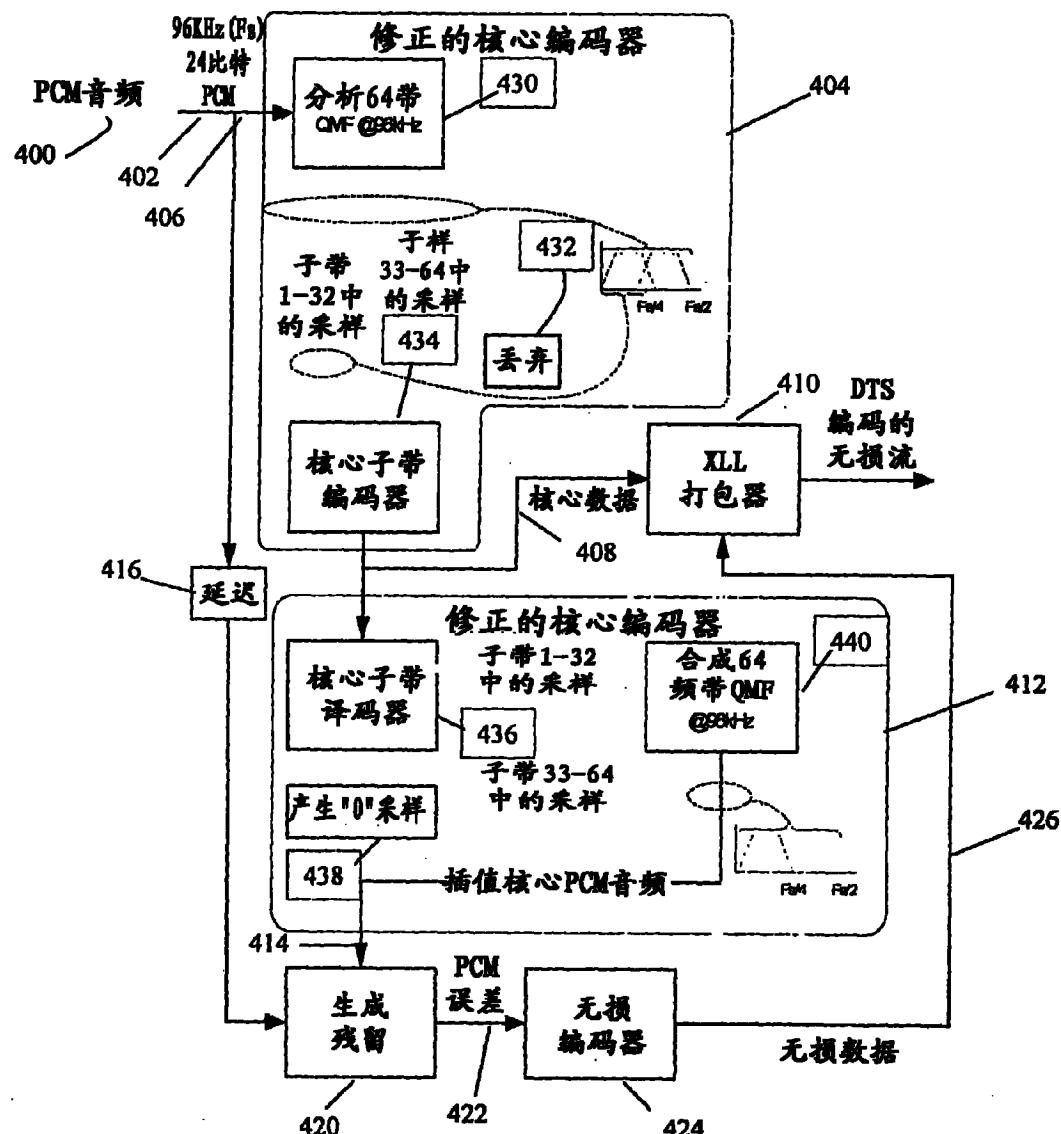


图 9