(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102667921 B
(45)授权公告日 2014.09.10

(21)申请号 201080058335.9
(22)申请日 2010.10.19
(30)优先权数据
61/253,459 2009.10.20 US
(85)PCT国际申请进入国家阶段日
2012.06.20
(86)PCT国际申请的申请数据
PCT/EP2010/065727 2010.10.19
(87)PCT国际申请的公布数据
WO2011/048100 EN 2011.04.28
(73)专利权人 弗兰霍菲尔运输应用研究公司
地址 德国慕尼黑
(72)发明人 纪尧姆·福奇斯
维内什·苏布巴拉曼
尼古劳斯·雷特尔巴赫
马库斯·穆赖特鲁斯 马克·伽依尔
帕特里克·瓦姆博尔德
克里斯蒂安·格里贝尔
奥利弗·魏斯
(74)专利代理机构 北京康信知识产权代理有限责任公司 11240
代理人 余刚 吴孟秋
(51)Int.Cl.
G10L 19/00(2013.01)
(56)对比文件
CN 101015216 A,2007.08.08,
CN 101160618 A,2008.04.09,
US 2005/0203731 A1,2005.09.15,
WO 2008/1501411 A1,2008.12.11,
Max Neuendorf等.A Novel Scheme for Low Bitrate Unified Speech and Audio Coding-MPEG RMO.《Audio Engineering Society 126th Convention Paper》.2009,1-13.
Eunju Imm等.Lossless coding of audio spectral coefficients using selective bitplane coding.《9th International Symposium on Communications and Information Technology》.2009,525-530.
审查员 陈宸

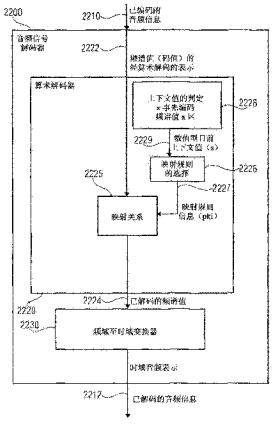
权利要求书4页 说明书33页 附图42页

(54)发明名称

音频编码器、音频解码器、用于将音频信息编码的方法、用于将音频信息解码的方法

(57)摘要

一种用以基于已编码的音频信息而提供已解码的音频信息的音频解码器(2200),其包含算术解码器(2200),用以基于频谱系数的经算术编码的表示(2222)提供多个已解码的频谱值(2224)。该音频解码器也包含频域至时域变换器(2230),用以使用该已解码的频谱值(2224)提供时域音频表示,来提供已解码的音频信息(2212)。该算术解码器被配置为依据描述目前上下文状态的数值型目前上下文值,而选定描述码值映射至符号码的映射规则。该算术解码器被配置为依据多个事先解码频谱值而判定数值型目前上下文值。该算术解码器被配置为使用迭代区间大小缩减而评估至少一个表来判定该数值型目前上下文值是否与由该表的一个登录项目所描述的表上下文值相同,或位于由该表的登录项目的描述的一区间内部,以及导出描述所选定的映射表的映射规则指数值。音频编码器也使用迭代区间表大小缩减。



CN 102667921 B

1. 一种用以基于已编码的音频信息（210 ;810 ;2210）而提供已解码的音频信息（212 ;812 ;2212）的音频解码器（200 ;800 ;2200），所述音频解码器包括 ：

算术解码器（230 ;820 ;2220），用以基于频谱值的经算术编码的表示（232 ;821 ;2222）而提供多个已解码的频谱值（232 ;822 ;2224）;及

频域至时域变换器（260 ;830 ;2230），用以使用所述已解码的频谱值（232 ;822 ;2224）提供时域音频表示，来获得所述已解码的音频信息（212 ;812 ;2212）;

其中，所述算术解码器（230 ;820 ;2220）被配置为依据描述目前上下文状态的数值型目前上下文值而选择描述算术式编码表示的一码值映射至表示所述已解码的频谱值中的一者或多者、或所述已解码的频谱值中的一者或多者的至少一部分的一符号码的映射规则（297），

其中，所述算术解码器被进一步配置为依据多个事先解码频谱值而判定所述数值型目前上下文值 ;

其中，所述算术解码器被进一步配置为使用迭代区间大小缩减（542 ;546）来评估至少一个表，而判定所述数值型目前上下文值是否与由所述表的一登录项目所描述的表上下文值相同，或是否位于所述表的登录项目所描述的一区间内部，以及导出描述已选择的映射规则的映射规则指数值。

2. 根据权利要求 1 所述的音频解码器（200 ;800 ;2200），其中，所述算术解码器（230 ;820 ;2220）被进一步配置为 ：

初始化一下区间边界变量而指定一初始表区间的下边界，

初始化一上区间边界变量而指定所述初始表区间的上边界，

评估一表登录项目，该项目的表指数设置于所述初始表区间的中心，而比较所述数值型目前上下文值与由所评估的表登录项目所表示的表上下文值，

依据该比较的结果而调整所述下区间边界变量或所述上区间边界变量，来获得一已更新的表区间，及

基于一个或多个已更新的表区间而重复一表登录项目的评估及所述下区间边界变量或所述上区间边界变量的调适，直至一表上下文值等于所述数值型目前上下文值，或直至由已更新的所述区间边界变量所界定的表区间大小达到或降至低于临界值表区间大小为止。

3. 根据权利要求 2 所述的音频解码器（200 ;800 ;2200），其中，所述算术解码器（230 ;820 ;2220）被进一步配置为响应于所述表的一给定的登录项目表示等于所述数值型目前上下文值的一表上下文值，而提供由所述表的给定的登录项目所描述的映射规则指数值。

4. 根据权利要求 1 所述的音频解码器（200 ;800 ;2200），其中，所述算术解码器（230 ;820 ;2220）被进一步配置为执行下列运算法则 ：

a）设定下区间边界变量 i_min 为 -1 ;

b）设定上区间边界变量 i_max 为表的登录项目数目减 1 ;

c）检查 i_max 与 i_min 间的差异是否大于 1，及重复下列步骤直至不再满足此项条件，或重复直至达到舍弃条件为止 ：

c1）设定变量 i 为 i_min+ ((i_max-i_min)/2)，

c2）若由具有表指数 i 的一表登录项目所描述的一表上下文值系大于所述数值型目前

上下文值，则设定上区间边界变量 i_max 为

i ；而若由具有表指数 i 的一表登录项目所描述的一表上下文值小于所述数值型目前上下文值，则设定下区间边界变量 i_min 为 i ；及

c3）若由具有表指数 i 的一表登录项目所描述的一表上下文值等于所述数值型目前上下文值，则舍弃 c)的重复，作为该运算法则的结果，返回由具有表指数 i 的所述表登录项目所描述的映射规则指数值。

5. 根据权利要求 1 所述的音频解码器（200 ；800 ；2200），其中，所述算术解码器被进一步配置为基于描述事先解码频谱值的振幅的振幅值的一加权组合而获得所述数值型目前上下文值。

6. 根据权利要求 1 所述的音频解码器（200 ；800 ；2200），其中，所述表包含多个登录项目，

其中，多个所述登录项目中的每一个描述一表上下文值及一相关联的映射规则指数值，及

其中，所述表的登录项目依据所述表上下文值而数值排序。

7. 根据权利要求 1 所述的音频解码器（200 ；800 ；2200），其中，所述表包含多个登录项目，

其中，多个所述登录项目中的每一个描述界定一上下文值区间的一边界值的一表上下文值，及与所述上下文值区间相关联的映射规则指数值。

8. 根据权利要求 1 所述的音频解码器（200 ；800 ；2200），其中，所述算术解码器（230 ；820 ；2220）被进一步配置为执行依赖于所述数值型目前上下文值的映射规则的二步骤式选择 ；

其中，所述算术解码器被进一步配置为在第一选择步骤（540）中检查所述数值型目前上下文值或自其中导出的值是否等于由一直接命中表的一登录项目所描述的一有效状态表上下文值 ；及

其中，所述算术解码器被进一步配置为在第二选择步骤（544）中判定在所述表的登录条目所描述的多个区间中，所述数值型目前上下文值位于由所述表的登录条目描述的哪个区间中，其中，所述第二选择步骤仅在所述数值型目前上下文值或自其中导出的值与所述直接命中表的登录项目所描述的所述有效状态表上下文值不同时执行 ；及

其中，所述算术解码器被进一步配置为使用所述迭代区间大小缩减（542 ；546）而评估所述直接命中表，而判定所述数值型目前上下文值是否与由所述直接命中表的一登录项目所描述的一有效状态表上下文值相同。

9. 根据权利要求 8 所述的音频解码器（200 ；800;2200），其中，所述算术解码器被进一步配置为在所述第二选择步骤（544）中使用迭代区间大小缩减（542 ；546），评估一区间映射表，该表的登录项目描述上下文值区间的边界值。

10. 根据权利要求 9 所述的音频解码器，其中，所述算术解码器（230 ；820;2220）被进一步配置为依据由登录项目所表示的区间边界表上下文值与所述数值型目前上下文值间的比较，而迭代地缩减一表区间的大小，直至所述表区间的大小达到或减至低于预定临界值表区间大小，或直至位于所述表区间中心的一表登录项目所描述的区间边界表上下文值等于所述数值型目前上下文值为止 ；及

其中，所述算术解码器被进一步配置为当所述表区间的大小迭代缩减被舍弃时，依据所述表区间的一区间边界设定值而提供所述映射规则指数值。

11. 一种用以基于输入的音频信息（110 ;710 ;2110）而提供已编码的音频信息（112 ;712 ;2112）的音频编码器（100 ;700 ;2100），所述音频解码器包括 ：

一能量压缩时域至频域变换器（130 ;720 ;2120），用以基于所述输入的音频信息的时域表示而提供一频域音频表示，使得所述频域音频表示（132 ;722 ;2124）包含一频谱值集合 ;及

一算术编码器（170 ;730 ;2130）被配置为使用一可变长度码字组而编码一频谱值或其预前处理版本，

其中，所述算术编码器（170;730;2130）被进一步配置为将一频谱值或一频谱值的最高有效位平面值映射至一码值，

其中，所述算术编码器被进一步配置为依据描述目前上下文状态的一数值型目前上下文值，而选择将一频谱值或一频谱值的最高有效位平面值映射至一码值的映射规则 ;及

其中，所述算术编码器被进一步配置为依据先前已编码的频谱值而判定所述数值型目前上下文值 ；

其中，所述算术编码器被进一步配置为使用一迭代区间大小缩减而评估至少一个表，来判定所述数值型目前上下文值是否与由所述表的登录项目所描述的一表上下文值相同，或是否位于由所述表的登录项目所描述的一区间内，且导出描述一所选定的映射规则的映射规则指数值。

12. 一种用以基于已编码的音频信息而提供已解码的音频信息的方法，所述方法包含 ：

基于频谱值的算术编码表示而提供多个已解码的频谱值 ;及

使用所述已解码的频谱值而提供一时域音频表示来获得所述已解码的音频信息 ；

其中，提供所述多个已解码的频谱值包括，依据描述一目前上下文状态的一数值型目前上下文值，而选择描述以编码形式表示一频谱值或一频谱值的最高有效位平面的一码值映射至，以解码形式表示一频谱值或一频谱值的最高有效位平面的一符号码映射规则 ;及

其中，所述数值型目前上下文值是依据多个事先解码频谱值而判定 ；

其中，至少一个表是使用一迭代区间大小缩减来评估，而判定所述数值型目前上下文值是否与由所述表的登录项目所描述的一表上下文值相同，或是否位于由所述表的登录项目所描述的一区间内部，且导出描述一所选定的映射规则的映射规则指数值。

13. 一种用以基于输入的音频信息而提供已编码的音频信息的方法，所述方法包含 ：

使用能量压缩的时域至频域变换，基于所述输入的音频信息的时域表示而提供一频域音频表示，使得所述频域音频表示包含频谱值集合 ;及

使用一可变长度码字组，算术式地编码一频谱值或其预处理版本，其中，一频谱值或一频谱值的最高有效位平面值被映射至一码值 ；

其中，描述一频谱值或一频谱值的最高有效位平面值的映射至一码值的映射规则是依据描述目前上下文状态的一数值型目前上下文值而选择 ；

其中，所述数值型目前上下文值是依据多个事先解码频谱值而判定 ;及

其中，至少一个表是使用一迭代区间大小缩减来评估，而判定所述数值型目前上下文

值是否与由所述表的登录项目所描述的一表上下文值相同，或是否位于由所述表的登录项目所描述的一区间内部，且判定描述一所选定的映射规则的映射规则指数值。

音频编码器、音频解码器、用于将音频信息编码的方法、用于将音频信息解码的方法

技术领域

[0001] 依据本发明的实施例是有关于一种用以基于已编码的音频信息而提供已解码的音频信息的音频解码器，一种用以基于输入的音频信息而提供已编码的音频信息的音频编码器，一种用以基于已编码的音频信息而提供已解码的音频信息的方法，一种用以基于输入的音频信息而提供已编码的音频信息的方法，及一种计算机程序。

[0002] 依据本发明的实施例是有关于一种改良式无噪声频谱编码，其可用于音频编码器或音频解码器，例如所谓的统一语音与音频编码器(USAC)。

背景技术

[0003] 后文中将简短解说本发明的背景，从而有助于了解本发明及其优点。过去十年间，大量努力致力于以良好位率效率而可能数字式储存与发布音频内容。此一方面有一项重大成就是国际标准 ISO/IEC14496-3 的定义。此一标准的第三部分是有关音频内容的编码及解码，而第三部分的第四子部分是有关一般音频编码。ISO/IEC14496 第三部分，第四子部分定义一般音频内容的编码及解码构想。此外，已提出进一步改良来改善质量和 / 或减低所要求的位率。

[0004] 依据该项标准所叙述的构想，时域音频信号被转换成时频表示。从时域变换成时频域典型地是使用时域样本的变换区块执行，该变换区块也称作为“帧”。已发现较佳是使用重叠帧，其移位例如半个帧，原因在于重叠允许有效地避免(或至少减少)假影(artifacts)。此外，已发现须进行开窗(windowing)，以免源自于此种时间上有限的帧处理的假影。

[0005] 通过将该输入的音频信号的一开窗部从时域变换成时频域，许多情况下，获得能量压缩，使得部分频谱值包含比多个其它频谱值显著更大的幅度。如此，许多情况下，幅度显著高于该等频谱值平均幅度的频谱值的数量相对较少。结果导致能量压缩的时域至时频域变换的一个典型例是所谓的修正离散余弦变换(MDCT)。

[0006] 频谱值经常是依据心理声学(psychoacoustic)模型而定标(scaled)及量化，使得针对心理声学上较重要的频谱值的量化误差较小，而针对心理声学上较不重要的频谱值的量化误差较大。已经定标与量化的频谱值被编码来提供其位率有效的表示。

[0007] 例如，所谓的量化频谱系数的霍夫曼编码的使用在国际标准 ISO/IEC14496-3:2005 (E)，第三部分，第四子部分中进行了描述。

[0008] 然而，已发现频谱值的编码质量对所要求的位率有显著影响。同样，已发现音频解码器的复杂程度是取决于用于编码该频谱值的编码处理，音频解码器经常制作成可携式消费者装置，因此须价廉且耗电量低。

[0009] 综上所述，需要可提供位率效率与计算量间的改良式折衷的一种音频内容的编码及解码构想。

发明内容

[0010] 依据本发明的一实施例，形成一种用以基于已编码的音频信息而提供已解码的音频信息的音频解码器。该音频解码器包含用以基于频谱系数的经算术编码的表示而提供多个已解码的频谱值的一算术解码器。该算术解码器也包含用以使用已解码的频谱值提供时域音频表示，来获得已解码的音频信息的频域至时域变换器。该算术解码器被配置为依据描述目前上下文状态的数值型目前上下文值而选择描述一码值映射至一符号码的映射规则。该算术解码器被配置为依据多个事先解码频谱值而判定该数值型目前上下文值。同样，该算术解码器被配置为使用迭代区间大小缩减来评估至少一个表，而判定该数值型目前上下文值是否与由该表的一登录项目所描述的表上下文值相同，或是否位于该表的登录项目所描述的一区间内部，来导出描述已选择的映射规则的映射规则指数值。

[0011] 依据本发明的一实施例是基于发现：可提供一数值型目前上下文值，该数值型目前上下文值描述用以解码一音频内容的频谱值的一算术解码器的一目前上下文状态，该数值型目前上下文值极为适合用以导出映射规则指数值，其中该映射规则指数值是使用基于一表的迭代区间大小，而描述要选用于该算术解码器的映射规则。已发现使用重复区间大小缩减的表搜寻极为适合依据一数值型目前上下文值，从相对较少的映射规则中选出映射规则(由映射规则指数值所描述)，其中该可能的映射规则的数目典型地比由该数值型目前上下文值所描述的可能上下文状态的数目更少，至少少十倍因子。详细分析已经显示适当映射规则的选择可通过使用迭代区间大小缩减而以高运算效率执行。即便在最恶劣情况下，表存取次数通过此构想可维持较小。当试图在实时环境下实施音频解码时，此点显示极为可能。此外，已发现迭代区间大小缩减可应用于检测一数值型目前上下文值是否与该表的一登录项目所描述的表上下文值相同，及应用于检测一数值型目前上下文值是否落在由该表的登录项目所描述的一区间内部。

[0012] 要言之，已发现使用迭代区间大小缩减极为适合用以执行哈希演绎法则，而依据一数值型目前上下文值，选定用于一音频内容的算术解码的映射规则，其中典型地该数值型目前上下文值的可能值的数目有效大于映射规则数目，来维持用于储存该等映射规则的有效地小量的内存要求。

[0013] 在一较佳实施例，该算术解码器被配置为初始设定一下区间边界变量而指定一初始表区间的下边界，及初始设定一上区间边界变量而指定一初始表区间的上边界。该算术解码器也较佳被配置为评估一表登录项目，其表指针排列于该初始表区间的中心，而比较计数与由该所评估的表的登录项目所表示的一表上下文值。该算术解码器也被配置为依据该比较结果而调适该下区间边界变量或上区间边界变量来获得一已更新的表区间。此外，该算术解码器被配置为基于一个或多个已更新的表区间而重复一表登录项目的评估及该下区间边界变量或该上区间边界变量的调适，直至一表上下文值等于该数值型目前上下文值，或直至由该已更新的区间边界变量所界定的表区间大小达到或降至低于临界值表区间大小为止。已发现重复区间大小缩减可使用前述各步骤而有效实施。

[0014] 在一较佳实施例，该算术解码器被配置为响应于该表的一给定的登录项目表示表上下文值，该值等于该数值型目前上下文值，而提供由该表的该给定的登录项目所描述的映射规则指针值。如此，实施极为有效的表存取机制，其极为适合硬件实施，原因在于典型地耗时且耗用电能的表存取次数维持于次数少。

[0015] 在一较佳实施例，该算术解码器被配置为执行下列运算法则，其中，在准备步骤，设定下区间边界变量 i_min 为 -1 ；及设定上区间边界变量 i_max 为表的登录项目数目减 1。在该运算法则，进一步检查 i_max 与 i_min 间的差异是否大于 1，及重复下列步骤直至不再符合此项条件（i_max-i_min>1），或重复直至达到舍弃条件为止：（1）设定变量 i 为 i_min+((i_max-i_min)/2)，（2）若由具有表指数 i 的表的登录项目所描述的表上下文值大于该数值型目前上下文值，则设定上区间边界变量 i_max 为 i，及（3）若由具有表指数 i 的表的登录项目所描述的表上下文值小于该数值型目前上下文值，则设定下区间边界变量 i_min 为 i。若由具有表指数 i 的表的登录项目所描述的表上下文值等于该数值型目前上下文值，则舍弃前述步骤（1）（2）（3）的重复。此种情况下，即，若由具有表指数 i 的表的登录项目所描述的表上下文值等于该数值型目前上下文值，则返回由具有该表指数 i 的该表的登录项目所描述的映射规则指针值。此种运算法则在音频解码器中的执行，提供选择映射规则时的极为良好的运算效率。

[0016] 在一较佳实施例，该算术解码器被配置为基于描述事先解码频谱值的振幅的振幅值的一加权组合而获得该数值型目前上下文值。已发现此项获得数值型目前上下文值的机制导致一数值型目前上下文值，其允许该迭代区间大小缩减来有效选择映射规则。其原因在于实际上，描述事先解码频谱值的振幅的振幅值的一加权组合导致一数值型目前上下文值，使得数值上相邻的数值型目前上下文值经常与目前要被解码的频谱值的类似的上下文环境有关。如此允许基于迭代区间大小缩减而有效应用哈希演绎法则。

[0017] 在一较佳实施例，该表包含多个登录项目，其中多个登录项目各自描述一表上下文值及相关联的映射规则指数值，及其中该表的登录项目是依据该些表上下文值而作数值排序。已发现此表极为适合用于组合该迭代区间大小缩减的应用。该表的登录项目的数值排序允许在相当少次重复次数以内，执行一表上下文值的搜寻，该表上下文值与该数值型目前上下文值相同，识别其中该数值型目前上下文值所在的区间。如此，表的存取次数维持少次。同样，通过将一表上下文值及相关联的映射规则组合在单一表登录项目内，可减少表的存取次数，其维持在硬件装置的执行时间短及其耗电量小。

[0018] 在一较佳实施例，该表包含多个登录项目，其中该些多个登录项目各自描述界定上下文值区间的一边界值的一表上下文值，及与该上下文值区间相关联的映射规则指数值。使用此种构想，可使用迭代区间大小缩减而有效识别该数值型目前上下文值所在的一区间。此外，可维持重复次数及表存取次数少。

[0019] 在一较佳实施例，该算术解码器被配置为依据该数值型目前上下文值而执行映射规则的二步骤式选择。此种情况下，该算术解码器被配置为在第一选择步骤，检查该数值型目前上下文值或自其中导出的值是否等于由直接命中表的一登录项目所描述的一有效状态值；该算术解码器被配置为在第二选择步骤，若该数值型目前上下文值或自其中导出的值与该直接命中表的该等登录项目所描述的该些有效状态值不同，则判定唯一执行哪一个映射规则，其中该数值型目前上下文值位于该些多个区间中的那个区间。该算术解码器被配置为该使用该迭代区间大小的缩减而评估该直接命中表，而判定该数值型目前上下文值是否与由该直接命中表的一登录项目所描述的一表上下文值相同。已发现通过使用此种二步骤式表评估机制，可有效地识别特别有效的上下文状态，该些特别有效的上下文状态是由该直接命中表的登录项目所描述，以及在该第二选择步骤，也可对较非有效的上下文状

态(其非由该直接命中表的登录项目所描述)选择适当的映射规则。通过此方式,可在第一选择步骤处理该最高有效的上下文状态,其减低在特别有效的状态存在下的运算复杂度。此外,即便对较非有效的状态仍可找到极为适合的映射规则。

[0020] 在一较佳实施例,该算术解码器被配置为在第二选择步骤,使用迭代区间大小缩减,评估一区间映射表,该表的登录项目描述上下文值区间的边界值。发现该迭代区间大小缩减极为适合用于直接命中的识别,及用于识别一数值型目前上下文值所在由该区间映射表所描述的多个区间中的哪一个区间。

[0021] 在一较佳实施例,该算术解码器被配置为依据由登录项目所表示的区间边界上下文值与该数值型目前上下文值间的比较,而重复地缩减一表区间的大小,直至一表区间的大小达到或减至低于预定临界值表区间大小,或直至位于该表区间中心的一表登录项目所描述的区间边界上下文值等于该数值型目前上下文值为止。该算术解码器被配置为当避免重复缩减该表区间的大小时,依据该表区间的一区间边界设定值而提供该映射规则指数值。使用此种构想,可以少量运算量而判定该数值型目前上下文值位于由该区间映射表的该些登录项目所界定的多个表区间中的哪一个表区间。如此,能够以低度运算量即可选定映射规则。

[0022] 依据本发明的一实施例,形成一种用以基于输入的音频信息而提供已编码的音频信息的音频编码器。该音频编码器包含用以基于该输入的音频信息的时域表示而提供一频域音频表示,使得该频域音频表示包含一频谱值集合的一能量压缩时域至频域变换器。该音频编码器也包含被配置为使用一可变长度码字组而编码一频谱值或其预处理版本的一算术编码器。该算术编码器被配置为将一频谱值或一频谱值的最高有效位平面值映射至一码值。该算术编码器被配置为依据描述目前上下文状态的一数值型目前上下文值,而选择将一频谱值或一频谱值的最高有效位平面值映射至一码值的映射规则。该算术编码器被配置为依据先前已编码的频谱值,而判定该数值型目前上下文值。该算术编码器被配置为使用一迭代区间大小缩减来评估至少一个表,而判定该数值型目前上下文值是否与由该表的登录项目所描述的一表上下文值相同,或是否位于由该表的登录项目所描述的一区间内部,以及由此导出描述一所选定的映射规则的映射规则指数值。此种音频信号编码器基于前文讨论的音频信号解码器的相同发现。已发现对音频内容的解码有效的映射规则的选择机制应该也适用于编码器端,来允许获得一致性系统。

[0023] 依据本发明的一实施例,形成一种用以基于已编码的音频信息而提供已解码的音频信息的方法。

[0024] 依据本发明的另一实施例,形成一种用以基于输入的音频信息而提供已编码的音频信息的方法。

[0025] 依据本发明的又一实施例,形成一种用于执行该些方法中的一个的计算机程序。

[0026] 该些方法及计算机程序是基于与前述音频解码器及前述音频编码器相同的发现。

[0027] 附图例明

[0028] 接着将参考附图描述依据本发明的实施例,附图中:

[0029] 图 1 显示依据本发明的一实施例的一种音频编码器的方块示意图;

[0030] 图 2 显示依据本发明的一实施例的一种音频解码器的方块示意图;

[0031] 图 3 显示用以解码频谱值的运算法则“value_decode ()”的虚拟程序码表示;

[0032] 图 4 显示用于状态计算的上下文的示意代表图；

[0033] 图 5a 显示用以映射上下文的运算法则“arith_map_context（）”的虚拟程序码表示；

[0034] 图 5b 及图 5c 显示用以获得上下文状态值的运算法则“arith_get_context（）”的虚拟程序码表示；

[0035] 图 5d 显示用以从状态变量导出累积－频率－表指数值“pki”的运算法则“get_pk（s）”的虚拟程序码表示；

[0036] 图 5e 显示用以从状态值导出累积－频率－表指数值“pki”的运算法则“arith_get_pk（s）”的虚拟程序码表示；

[0037] 图 5f 显示用以从状态值导出累积－频率－表指示值“pki”的运算法则“get_pk（unsigned long s）”的虚拟程序码表示；

[0038] 图 5g 显示用以从可变长度码字组算术地解码一符号的运算法则“arith_decode（）”的虚拟程序码表示；

[0039] 图 5h 显示用以更新上下文的运算法则“arith_update_context（）”的虚拟程序码表示；

[0040] 图 5i 显示定义及变量的图例；

[0041] 图 6a 显示统一语音与音频编码器（USAC）原始数据区块的语法表示；

[0042] 图 6b 显示单一信道元素的语法表示；

[0043] 图 6c 显示成对信道元素的语法表示；

[0044] 图 6d 显示“ics”控制信息的语法表示；

[0045] 图 6e 显示频域信道串流的语法表示；

[0046] 图 6f 显示算术式编码频谱数据的语法表示；

[0047] 图 6g 显示解码一频谱值集合的语法表示；

[0048] 图 6h 显示数据元素及变量的图例；

[0049] 图 7 显示依据本发明的另一实施例的一种音频编码器的方块示意图；

[0050] 图 8 显示依据本发明的另一实施例的一种音频解码器的方块示意图；

[0051] 图 9 显示使用依据本发明的编码方案，依据 USAC 草拟标准的工作草案 3，用于无噪声编码比较的配置；

[0052] 图 10a 显示用于状态计算的上下文当其用于依据 USAC 草拟标准的工作草案 4 时的示意代表图；

[0053] 图 10b 显示用于状态计算的上下文当其用于依据本发明的实施例时的示意代表图；

[0054] 图 11a 显示该表当其用于依据 USAC 草拟标准的工作草案 4 的该算术编码方案时的综论；

[0055] 图 11b 显示该表当其用于依据本发明的算术编码方案时的综论；

[0056] 图 12a 显示用于依据本发明及依据 USAC 草拟标准的工作草案 4 的无噪声编码方案的只读存储器需求指令的图解代表图；

[0057] 图 12b 显示依据本发明及依据 USAC 草拟标准的工作草案 4 的构想的总 USAC 解码器数据只读存储器需求指令的图解代表图；

[0058] 图 13a 显示使用依据 USAC 草拟标准的工作草案 3 的算术编码器、及依据本发明的一实施例的算术解码器，统一语音与音频编码编码器所使用的平均位率的表代表图；

[0059] 图 13b 显示使用依据 USAC 草拟标准的工作草案 3 的算术编码器、及依据本发明的一实施例的算术编码器，用于统一语音与音频编码编码器的位累积控制的表代表图；

[0060] 图 14 显示依据 USAC 草拟标准的工作草案 3、及依据本发明的一实施例，用于 USAC 编码器的平均位率的表代表图；

[0061] 图 15 显示按照帧的 USAC 的最小、最大、及平均位率的表代表图；

[0062] 图 16 显示按照帧的最佳状况及最恶劣状况的表代表图；

[0063] 图 17（1）及图 17（2）显示表“ari_s_hash[387]”的内容的表代表图；

[0064] 图 18 显示表“ari_gs_hash[225]”的内容的表代表图；

[0065] 图 19（1）及图 19（2）显示表“ari_cf_m[64][9]”的内容的表代表图；以及

[0066] 图 20（1）及图 20（2）显示表“ari_s_hash[387]”的内容的表代表图；以及

[0067] 图 21 显示依据本发明的一实施例的一种音频编码器的方块示意图；及

[0068] 图 22 显示依据本发明的一实施例的一种音频解码器的方块示意图。

具体实施方式

[0069] 1. 依据图 7 的音频编码器

[0070] 图 7 显示依据本发明的一实施例的一种音频编码器的方块示意图。音频编码器 700 被配置为接收输入的音频信息 710，并基于此而提供已编码的音频信息 712。音频编码器包含能量压缩时域至频域变换器 720，该变换器被配置为基于该输入的音频信息 710 的时域表示而提供频域音频表示 722，使得该频域音频表示 722 包含一频谱值集合。音频编码器 700 也包含算术编码器 730，该算数编码器被配置为使用一可变长度码字组而编码（形成该频域音频表示 722 的频谱值集合中的）一频谱值或其预处理版本，来获得已编码的音频信息 712（其可包含例如多数可变长度码字组）。

[0071] 算术编码器 730 被配置为依据上下文状态，而将一频谱值或频谱值的一最高有效位平面值映射至一码值（即，映射至一可变长度码字组）。算术编码器 730 被配置为依据上下文状态，选择描述将一频谱值或频谱值的一最高有效位平面值映射至一码值的映射规则。算术编码器被配置为依据多个事先编码的（优选但非必要地，相邻的）频谱值而判定该目前上下文状态。为了达成此目的，算术编码器被配置为检测一组多个事先编码相邻频谱值（其是单独地或共同地满足有关其幅度的预定状况），并依据该检测结果而判定该目前上下文状态。

[0072] 如此可知，一频谱值或频谱值的一最高有效位平面值映射至一码值可通过使用映射规则 742 由频谱值编码 740 执行。状态追踪器 750 可被配置为追踪该上下文状态，且可包含一群组检测器 752 来检测一组多个事先编码相邻频谱值（其是单独地或共同地满足有关其幅度的预定状况）。状态追踪器 750 也较佳被配置为依据由该群组检测器 752 所执行的该检测结果而判定目前上下文状态。如此，状态追踪器 750 提供描述该目前上下文状态的信息 754。映射规则选择器 760 可选择映射规则，例如累积频率表，其描述一频谱值或频谱值的一最高有效位平面值映射至一码值。如此，映射规则选择器 760 将映射规则信息 742 提供至该频谱编码 740。

[0073] 综上所述，音频编码器 700 执行由该时域至频域变换器所提供的一频域音频表示的算术编码。该算术编码为上下文相依性，使得映射规则（例如累积频率表）是依据事先编码频谱值而选择。如此，时间和 / 或频率（或至少在预定环境内）是彼此相邻和 / 或与该目前编码频谱值（即，在该目前编码频谱值的预定环境内的频谱值）相邻的频谱值在算术编码中被考虑从而调整由该算术编码评估的概率分布。当选定适当的映射规则时，执行检测来测定是否有一组多个事先编码相邻频谱值是单独地或共同地满足有关其幅度的预定状况。此项检测结果是应用于该目前上下文状态的选择，即，应用在映射规则的选择。通过检测是否有一组多数频谱值其是特小或特大，可辨识频域音频表示（其可为时频表示）内的特定特征。该特定特征（诸如一组多数特小的或特大的频谱值）指示应当使用的特定上下文状态，原因在于此一特定上下文状态可提供极佳编码效率。如此，检测满足预定状况的该组相邻频谱值，该检测通常是用来与基于多个事先编码频谱值的一组合的可替换上下文评估相结合地使用，提供一种机制，其允许有效地选定适当的上下文，该输入的音频信息是否具有某些特殊状态（例如，包含大的被遮蔽的频率范围）。

[0074] 如此，可达成有效编码，同时维持上下文的计算充分简单。

[0075] 2. 依据图 8 的音频解码器

[0076] 图 8 显示音频解码器 800 的方块示意图。音频解码器 800 被配置为接收已编码的音频信息 810，并基于此而提供已解码的音频信息 812。音频解码器 800 包含算术解码器 820，该算数解码器被配置为基于频谱值的算术式编码表示 821 而提供多个已解码的频谱值 822。音频解码器 800 也包含频域至时域变换器 830，该变换器被配置为接收已解码的频谱值 822，并使用该已解码的频谱值 822，提供时域音频表示 812（其可组成该已解码的音频信息），来获得已解码的音频信息 812。

[0077] 算术解码器 820 包含频谱值测定器 824，该测定器被配置为将算术式编码的频谱值表示的一码值映射至表示已解码的频谱值中的一者或多者、或已解码的频谱值中的一者或多者的至少一部分（例如，最高有效位平面）的一符号码。频谱值测定器 824 可被配置为依据映射规则而执行映射，该映射规则可由映射规则信息 828a 描述。

[0078] 算术解码器 820 被配置为依据上下文状态（其可由上下文状态信息 826a 描述），选择描述一码值（由算术式编码的频谱值表示 821 描述）映射至一符号码（描述一个或多个频谱值）的映射规则。算术解码器 820 被配置为依据多数事先解码频谱值 822 而判定该目前上下文状态。为了达成此目的，可使用状态追踪器 826，其接收描述事先解码频谱值的信息。算术解码器也被配置为检测一组多个事先解码（优选但非必要地，相邻的）的频谱值（其是单独地或共同地满足有关其幅度的预定状况），并依据该检测结果而判定该目前上下文状态（例如，由上下文状态信息 826a 描述）。

[0079] 检测满足有关其幅度的预定状况的该组多个事先解码相邻频谱值例如可由一群组检测器（其是状态追踪器 826 的一部分）而进行。如此，获得目前上下文状态信息 826a。该映射规则的选择可由映射规则选择器 828 执行，该映射规则选择器从该目前上下文状态信息 826a 中导出映射规则信息 828a，并且将该映射规则信息 828a 提供至该频谱值测定器 824。

[0080] 有关该音频信号解码器 800 的功能，须注意该算术解码器 820 被配置为选择平均地极为适合用于要被解码的频谱值的映射规则（例如累积频率表），原因在于该映射规则是

依据目前上下文状态而选定，而该目前上下文状态又是依据多个事先解码频谱值而判定。如此，可利用要被解码的相邻频谱值间的统计相依性。此外，通过检测一组多个事先解码相邻频谱值其是单独地或共同地满足有关其幅度的预定状况，可调整映射规则适应事先解码频谱值的特殊状况(或样式)。例如，若识别一组多个较小的事先解码相邻频谱值，或若识别一组多个较大的事先解码相邻频谱值，则可选出特定映射规则。已发现存在有一组较大频谱值、或存在有一组较小频谱值可被视为须使用特别适用于此种状况的一专用映射规则的显著指示。如此，通过利用此组多个频谱值的检测可协助(或加速)上下文运算。同样，若未应用前述构想，则一音频内容的特性可视为不容易考虑。例如，比较用于正常上下文运算的该频谱值集合，一组多个事先解码频谱值其是单独地或共同地满足有关其幅度的预定状况的检测可基于不同的一频谱值集合执行。

[0081] 进一步细节稍后详述。

[0082] 3. 依据图 1 的音频编码器

[0083] 后文中，将叙述依据本发明的一实施例的音频编码器。图 1 显示此种音频编码器 100 的方块示意图。

[0084] 音频编码器 100 被配置为接收一输入的音频信息 110，及基于此提供一位串流 112，其构成一已编码的音频信息。音频编码器 100 选择性地包含一预处理器 120，其被配置为接收该输入的音频信息 110，及基于此而提供预处理输入的音频信息 110a。音频编码器 100 也包含一能量压缩时域至频域信号变换器 130，其也定名为信号变换器。信号变换器 130 被配置为接收输入的音频信息 110、110a，及基于此而提供一频域音频信息 132，其较佳是呈一频谱值集合形式。例如，信号变换器 130 被配置为接收输入的音频信息 110、110a 的一帧(例如时域样本的一区块)，及提供表示该个别音频帧的音频内容的一频谱值集合。此外，该信号变换器 130 可被配置为接收多个接续的、重叠或非重叠输入的音频信息 110、110a 的音频帧，及基于此而提供一时频域音频表示，其包含与各帧相邻频谱值接续频谱值集合的一序列，亦即一个频谱值集合。

[0085] 能量压缩时域至频域信号变换器 130 可包含一能量压缩滤波器排组，其是提供与不同的、重叠或非重叠频率范围相关联的频谱值。例如，该信号变换器 130 可包含一开窗 MDCT 变换器 130a，其被配置为使用一变换窗而开窗该输入的音频信息 110、110a(或其帧)，及执行该开窗输入的音频信息 110、110a (或其开窗帧) 的修正离散余弦变换。如此，该频域音频表示 132 可包含与该输入的音频信息的一帧相关联的呈 MDCT 系数形式的例如 1024 个频谱值的一集合。

[0086] 音频编码器 100 可选择性地进一步包含一频谱后处理器 140，其被配置为接收频域音频表示 132，及基于此而提供一后处理频域音频表示 142。该频谱后处理器 140 例如可被配置为执行时间噪声成形、和 / 或长期预测、和 / 或本领域已知的任何其它频谱后处理。音频编码器选择性地进一步包含定标器 / 量化器 150，其被配置为接收频域音频表示 132 或其后处理版本 142，及提供一已定标且已量化的频域音频表示 152。

[0087] 音频编码器 100 选择性地，进一步包含一心理声学模型处理器 160，其被配置为提供该输入的音频信息 110 (或其的后处理版本 110a)，及基于此而提供一选择性控制信息，其可用于能量压缩时域至频域信号变换器 130 的控制，用于选择性的频谱后处理器 140 的控制，和 / 或用于选择性的定标器 / 量化器 150 的控制。举例而言，心理声学模型处理器

160 可被配置为分析该输入的音频信息，判定该输入的音频信息 110、110a 的哪些分量对于人类的音频内容听觉特别重要，而该输入的音频信息 110、110a 的哪些分量对于人类的音频内容听觉较不重要。据此，心理声学模型处理器 160 可提供控制信息，其是由音频编码器 100 使用来调整由该定标器 / 量化器 150 对频域音频表示 132、142 的定标、和 / 或由该定标器 / 量化器 150 所施加的量化分辨率。结果，听觉上重要的标度因子频带（即，对人类的音频内容听觉特别重要的相邻频谱值组）是以大的定标因子定标且以相对较高分辨率量化，听觉上较不重要的标度因子频带（即，成组的相邻频谱值）是以较小的定标因子定标且以较低分辨率量化。据此，典型地，听觉上较为重要的频率的已定标频谱值明显大于听觉上较不重要的频谱值。

[0088] 音频编码器也包含一算术编码器 170，其被配置为接收频域音频表示 132（或者，可替换地，该频域音频表示 132 的后处理版本 142，或甚至该频域音频表示 132 本身）的已定标且已量化版本 152，及基于此而提供算术码字组信息 172a，使得该算术码字组信息表示该频域音频表示 152。

[0089] 音频编码器 100 也包含位串流有效负载格式化器 190，其被配置为接收该算术码字组信息 172a。该位串流有效负载格式化器 190 也典型地被配置为接收额外信息，例如描述哪些标度因子已经被定标器 / 量化器 150 应用的标度因子信息。此外，位串流有效负载格式化器 190 可被配置为接收其它控制信息。位串流有效负载格式化器 190 被配置为基于所接收的信息，通过依据期望的位串流语法而组装该位串流来提供该位串流 112，稍后详述。

[0090] 后文中，将叙述有关算术编码器 170 的细节。算术编码器 170 被配置为接收该频域音频表示 132 的多个后处理且已定标且已量化的频谱值。算术编码器包含一最高有效位平面提取器 174，其被配置为从一频谱值提取最高有效位平面 m。此处须注意，最高有效位平面可包含一个或甚至多个位（例如 2 或 3 位）其是该频谱值的最高有效位。如此，最高有效位平面提取器 174 提供一频谱值的最高有效位平面值 176。

[0091] 算术编码器 170 也包含一第一码字组测定器 180，其被配置为测定表示该最高有效位平面值 m 的算术码字组 acod_m[pki][m]。选择性地，码字组测定器 180 也提供一个或多个逸出码字组（此处也标示以“ARITH_ESCAPE”），指示例如多少个较低有效位平面（以及结果，指示该最高有效位平面的数值型权重）可用。第一码字组测定器 180 可被配置为使用具有（或参考）累积频率表指数 pki 的一选定的累积频率表而提供与最高有效位平面值 m 相关联的该码字组。

[0092] 为了判定是否应选择该累积频率表，该算术编码器较佳包含一状态追踪器 182，其被配置为例如通过观察哪些频谱值是事先编码而追踪该算术编码器的状态。结果，该状态追踪器 182 提供一状态信息 184，例如以“s”或“t”标示的状态值。算术编码器 170 也包含一累积频率表选择器 186，其被配置为接收该状态信息 184，并将描述该选定的累积频率表的信息 188 提供给该码字组测定器 180。举例而言，累积频率表选择器 186 可提供一累积频率表指数“pki”描述 64 累积频率表的一集合中哪一个累积频率表被选择用于由该码字组测定器使用。可替换地，累积频率表选择器 186 可将整个选定的累积频率表提供给该码字组测定器。如此，码字组测定器 180 可使用所择定的累积频率表来提供该最高有效位平面值 m 的码字组 acod_m[pki][m]，使得编码该最高有效位平面值 m 的实际码字组 acod_m[pki][m] 是与 m 值及累积频率表指数 pki 有相依性，并因此与该目前状态信息 184 有相依

性。有关编码处理及所获得码字组格式的进一步细节稍后详述。

[0093] 算术编码器 170 又包含一较低有效位平面提取器 189a，其被配置为如果要被解码的频谱值中的一者或多者超过只使用该最高有效位平面所能编码的数值范围，则从该已定标且已量化的频域音频表示 152 提取一个或多个较低有效位平面。若有所需，该等较低有效位平面可包含一个或多个位。据此，该较低有效位平面提取器 189a 提供较低有效位平面信息 189b。算术编码器 170 也包含一第二码字组测定器 189c，其被配置为接收较低有效位平面信息 189d，及基于此而提供表示 0、1、或更多较低有效位平面的内容的 0、1、或更多个码字组“acor_r”。该第二码字组测定器 189c 可被配置为应用算术编码运算法则或任何其它编码运算法则，而从该较低有效位平面信息 189b 导出该些较低有效位平面码字组“acor_r”。

[0094] 此处须注意，较低有效位平面数目可取决于该些已定标且已量化的频谱值 152 而改变，使得如果要被编码的已定标且已量化的频谱值为较小，则可能根本没有较低有效位平面；使得如果要被编码的该目前已定标且已量化的频谱值为中等范围，则可有一个较低有效位平面；及使得如果要被编码的已定标且已量化的频谱值具有较大值，则可有多于一个较低有效位平面。

[0095] 综上所述，算术编码器 170 被配置为使用阶层编码处理而编码已定标且已量化的频谱值，其是由该信息 152 描述。最高有效位平面(例如，每个频谱值包含 1、2 或 3 位)被编码来获得最高有效位平面值的一算术码字组“acod_m[pki][m]”。一个或多个较低有效位平面(该些较低有效位平面各自例如包含 1、2 或 3 位) 被编码从而获得一个或多个码字组“acod_r”。当编码最高有效位平面时，该最高有效位平面的值 m 被映射至一码字组 acod_m[pki][m]。为了达成此目的，64 个不同累积频率表是可用的，用于依据算术编码器 170 的状态，亦即依据事先编码频谱值来编码值 m。如此，获得码字组“acod_m[pki][m]”。此外，若存在有一个或多个较低有效位平面，则提供一个或多个码字组“acod_r”且包含至该位串流。

[0096] 重置描述

[0097] 音频编码器 100 选择性地可被配置为判定经由重置该内容，例如经由将该状态指标重置至一默认值，是否可获得位率的改良。如此，音频编码器 100 可被配置为提供一重置信息(例如，定名“arith_reset_flag”)，指示该算术编码内容是否经重置，及也指示于相对应解码器中用于算术解码的内容是否应重置。

[0098] 有关位串流格式及应用的累积频率表的细节稍后详述。

[0099] 4. 音频解码器

[0100] 后文中，将叙述依据本发明的一实施例的音频解码器。图 2 显示此种音频解码器 200 的方块示意图。

[0101] 音频解码器 200 被配置为接收一位串流 210，其表示一已编码的音频信息，并且其可与由音频编码器 100 所提供的位串流 112 相同。音频解码器 200 基于该位串流 210 而提供已解码的音频信息 212。

[0102] 音频解码器 200 包含一选择性的位串流有效负载解格式化器 220，其被配置为接收该位串流 210，并且从该位串流 210 提取一已编码的频域音频表示 222。例如，该位串流有效负载解格式化器 220 可被配置为从位串流 210，提取算术式编码的频谱值，例如表示该

频域音频表示的频谱值 a 的最高有效位平面值 m 的一算术码字组“acod_m[pki][m]”，及表示该频谱值 a 的较低有效位平面的内容的码字组“acod_r”。如此，已编码的频域音频表示 222 组成(或包含)频谱值的一算术式编码表示。该位串流有效负载解格式化器 220 进一步被配置为从该位串流提取额外控制信息，其未显示于图 2。此外，位串流有效负载解格式化器选择性地被配置为从位串流 210 提取一状态重置信息 224，其也被标示为算术重置标记或“arith_reset_flag”。

[0103] 音频解码器 200 包含一算术解码器 230，其也称作为“频谱无噪声解码器”。算术解码器 230 被配置为接收该已编码的频域音频表示 220，及选择性地，接收状态重置信息 224。算术解码器 230 也被配置为提供一已解码的频域音频表示 232，其可包含已解码的频谱值表示。举例而言，已解码的频域音频表示 232 可包含已解码的频谱值表示，其是由已编码的频域音频表示 220 描述。

[0104] 音频解码器 200 也包含一可选的反量化器 / 复位蔡器 240，其被配置为接收该已解码的频域音频表示 232，及基于此而提供已反量化及已复位标的频域音频表示 242。

[0105] 音频解码器 200 进一步包含一可选的频谱预处理器 250，其被配置为接收该已反量化及已复位标的频域音频表示 242，及基于此而提供该已反量化及已复位标的频域音频表示 242 的一预处理版本 252。音频解码器 200 也包含一频域至时域信号变换器 260，其也称作“信号变换器”。信号变换器 260 被配置为接收该已反量化及已复位标的频域音频表示 242 的该域处理版本 252（或者，可替换地，该已反量化及已复位标的频域音频表示 242 或已解码的频域音频表示 232），及基于此而提供该音频信息的一时域表示 262。该频域至时域信号变换器 260 例如可包含用以执行修正离散余弦反变换（IMDCT）及适当开窗(以及其它辅助功能，例如重叠与相加）的一变换器。

[0106] 音频解码器 200 进一步可包含可选的时域后处理器 270，其被配置为接收音频信息的时域表示 262，及使用时域后处理而获得已解码的音频信息 212。但若省略该后处理，则时域表示 262 可与已解码的音频信息 212 相同。

[0107] 此处须注意，反量化器 / 复位标器 240、频谱预处理器 250、频域至时域信号变换器 260、及时域后处理器 270 可依据控制信息加以控制，该控制信息是由位串流有效负载解格式化器 220 而提取自该位串流 210。

[0108] 总而言之，音频解码器 200 的整体功能、已解码的频域音频表示 232（例如与已编码的音频信息的音频帧相关联的的一频谱值集合），可使用算术解码器 230 基于已编码的频域音频表示 222 而获得。结果，例如 1024 个频谱值（其可为 MDCT 系数）的集合是经反量化、经复位标、及经预处理。如此，获得已经反量化、经复位标、及经频谱预处理的频谱值集合(例如，1024 个 MDCT 系数)。随后，自该已经反量化、经复位标、及经频谱预处理的频谱值集合(例如，MDCT 系数）而导出一音频帧的时域表示。如此，获得一音频帧的时域表示。一给定音频帧的时域表示可组合先前和 / 或后续音频帧的时域表示。举例而言，可执行后续音频帧的时域表示间的重叠及相加，从而平滑相邻音频帧的时域表示之间的变迁，并且获得频迭消除(aliasing cancellation)。有关基于已解码的频域音频表示 232 而重构已解码的音频信息 212 的相关细节例如可参考国际标准 ISO/IEC14496-3，部分 3，子部分 4 的详细讨论。但也可使用其它更精细的重叠及频迭消除方案。

[0109] 后文中，将叙述有关算术解码器 230 的若干细节。算术解码器 230 包含最高有效位

平面测定器 284，其被配置为接收描述最高有效位平面值 m 的算术码字组 acod_m[pki][m]。最高有效位平面测定器 284 可被配置为使用一包含多个 64 累积频率表集合中的一个累积频率表用以自该算术码字组“acod_m[pki][m]”而导出最高有效位平面值 m。

[0110] 最高有效位平面测定器 284 被配置为基于码字组 acod_m 而导出频谱值的一最高有效位平面的值 286。算术解码器 230 进一步包含较低有效位平面测定器 288，其被配置为接收表示一频谱值的一个或多个较低有效位平面的一个或多个码字组“acod_r”。如此，较低有效位平面测定器 288 被配置为提供一个或多个较低有效位平面的解码值 290。音频解码器 200 也包含一位平面组合器 292，其被配置为接收该些频谱值的最高有效位平面的解码值 286；并且如果这种较低有效位平面对于目前频谱值可用，则可接收该些频谱值的一个或多个较低有效位平面的解码值 290。如此，位平面组合器 292 提供已解码的频谱值，其是该已解码的频域音频表示 232 的一部分。当然，算术解码器 230 典型地被配置为提供多个频谱值，从而获得与该音频内容的目前帧相关联的已解码的频谱值的一全集。

[0111] 算术解码器 230 进一步包含一累积频率表选择器 296，其被配置为依据描述该算术解码器状态的一状态指标 298 而选择 64 个累积频率表中的一个。算术解码器 230 进一步包含一状态追踪器 299，其被配置为依据事先解码频谱值而追踪算术解码器的状态。该状态信息可选择性地响应于状态重置信息 224 而被重置为一默认状态信息。如此，累积频率表选择器 296 被配置为提供选定的累积频率表的指数（例如 pki）、或累积频率表本身用来依据码字组“acod_m”而应用于最高有效位平面值 m 的解码。

[0112] 概述音频解码器 200 的功能，音频解码器 200 被配置为接收一经位率有效地编码的频域音频表示 222，及基于此而获得已解码的频域音频表示。在用来基于已编码的频域音频表示 222 而获得已解码的频域音频表示 232 的算术解码器 230 中，通过使用算术解码器 280（其被配置为应用累积频率表）而开发相邻频谱值的最高有效位平面值间的不同组合的概率。换言之，通过依据状态指标 298（其是通过观察事先运算解码频谱值而得的）而从包含 64 个不同累积频率表的一集合中选出不同的累积频率表，来开发频谱值间的统计相依性。

[0113] 5. 频谱无噪声编码工具的综论

[0114] 后文中，将解说有关由例如算术编码器 170 及算术解码器 230 执行的编码及解码运算法则的细节。

[0115] 重点是放在解码运算法则的说明。但须注意，相对应的编码运算法则可依据解码运算法则的教导执行，其中映射是逆向的。

[0116] 须注意，后文将讨论的解码是用来允许典型地经后处理典型地经后处理、经定标且经量化的频谱值的所谓的“频谱无噪声编码”。频谱无噪声编码是用在音频编码 / 解码构想来进一步降低量化频谱的冗余，该量化频谱是例如经由能量压缩时域至频域变换器获得。

[0117] 用于本发明的实施例的频谱无噪声编码方案是基于算术编码结合动态调适上下文。频谱无噪声编码被馈以量化频谱值（其原始表示或已编码表示），并使用例如从多个事先解码邻近频谱值中导出的上下文相依性累积频率表。此处，时间上及频率上这二者的邻近皆列入考虑，如图 4 所示。然后，累积频率表（稍后详述）由算术编码器用来产生一可变长度二进制码，并由算术解码器用来从一可变长度二进制码导出解码值。

[0118] 举例而言，算术编码器 170 依据各个概率，对一给定符号集合产生二进制码。该二进制码是经由将该符号集合所在的一概率区间映射至一码字组而产生。

[0119] 后文中，将提供频谱无噪声编码工具的另一项短综论。频谱无噪声编码是用来进一步缩减量化频谱的冗余。该频谱无噪声编码方案是基于算术编码结合动态调适上下文。无噪声编码被馈以量化频谱值，并使用例如从七个事先解码邻近频谱值中导出的上下文相依性累积频率表。

[0120] 此处，时间上及频率上二者的邻近皆列入考虑，如图 4 所示。然后，累积频率表由算术编码器用来产生一可变长度二进制码。

[0121] 算术编码器对一给定符号集合及其各个概率产生二进制码。该二进制码是经由将该符号集合所在的一概率区间映射至一码字组而产生。

[0122] 6. 解码程序

[0123] 6.1. 解码处理综论

[0124] 后文中，将参考图 3 给予解码频谱值的程序的综合讨论，该图显示解码多个频谱值的程序的伪程序码表示。

[0125] 解码多个频谱值的程序包含上下文的初始化 310。上下文的初始化 310 包含使用函数“arith_map_context (lg)”从前一个上下文导出该目前上下文。从前一个上下文导出该目前上下文可包含该上下文的重置。上下文的重置及从前一个上下文导出该目前上下文这二者稍后详述。

[0126] 多个频谱值的解码也包含频谱值解码 312 及上下文更新 314 的迭代，该上下文更新是由函数“Arith_update_context (a,I,lg)”执行，稍后详述。频谱解码 312 及上下文更新 314 被重复 lg 次，其中 lg 是指示(例如，针对一音频帧) 要被解码的频谱值数目。频谱值解码 312 包含上下文值计算 312a、最高有效位平面解码 312b、及较低有效位平面加法 312c。

[0127] 状态值运算 312a 包括使用函数“arith_get_context (I,lg,arith_reset_flag,N/2)”运算第一状态值 s，该函数返回该第一状态值 s。该状态值运算 312a 也包含位准值“lev0”及位准值“lev”的运算，该些位准值“lev0”、“lev”是通过将第一状态值 s 向右位移 24 位获得的。该状态值运算 312a 也包含依据图 3 显示在参考标号 312a 的公式，运算第二状态值 t。

[0128] 最高有效位平面解码 312b 包含解码运算法则 312ba 的迭代执行，其中初次执行运算法则 312ba 之前，变量 j 被初始化为 0。

[0129] 运算法则 312ba 包含使用函数“arith_get_pk ()”，依据第二状态值 t，并且也依据位准值“lev”及 lev0 运算状态指数“pki”(也用作为累积频率表指数)，稍后后详述。运算法则 312ba 也包含依据状态指数 pki 而选择累积频率表，其中变量“cum_freq”可依据状态指数 pki 而设定至 64 累积频率表中的一个起始地址。同样，变量“cfl”可被初始化为所选定的累积频率表长度，其(例如) 等于字母表中的符号数目，即，可解码的不同值的数目。从“arith_cf_m[pki=0][9]”至“arith_cf_m[pki=63][9]”的全部累积频率表中可用于最高有效位平面值 m 解码的长度为 9，原因在于 8 个不同最高有效位平面值及一个逸出符号可被解码。随后考虑所选的累积频率表(由变量“cum_freq”及变量“cfl”描述)，通过执行函数“arith_decode ()”可获得最高有效位平面值 m。当导出最高有效位平面值 m 时，可评估

位串流 210 中名为“acod_m”的位(例如参考图 6g)。

[0130] 运算法则 312ba 也包含检验最高有效位平面值 m 是否等于逸出符号“ARITH_ESCAPE”。若最高有效位平面值 m 不等于该算术逸出符号,则舍弃运算法则 312ba (“断裂”-状况),因而运算法则 312ba 的其余指令被跳过。如此,该处理程序的执行是以设定频谱值 a 为等于最高有效位平面值 m 来继续(指令“a=m”)。相反地,若最高有效位平面值 m 是与算术逸出符号“ARITH_ESCAPE”相等,则位准值“lev”递增 1。如所述,然后重复运算法则 312ba 直至解码的最高有效位平面值 m 不同于该算术逸出符号为止。

[0131] 一旦完成最高有效位平面解码,即已经解码与该算术逸出符号不同的最高有效位平面值 m,则频谱值变量“a”设定为等于最高有效位平面值 m。随后,获得较低有效位平面,例如如图 3 以参考标号 312c 所示。针对该频谱值的各个较低有效位平面,解码两个二进制值中的一个。举例而言,获得较低有效位平面值 r。随后,通过将频谱值变数“a”向左位移 1 位,及通过加上目前解码的较低有效位平面值 r 作为最低有效位,而更新频谱值变量“a”。但须注意,本发明并未特别推荐获得较低有效位平面的构想。在某些情况下,甚至可省略任何较低有效位平面的解码。可替换地,可使用不同解码运算法则用于达成此目的。

[0132] 6.2. 依据图 4 的解码顺序

[0133] 后文中,将叙述频谱值的解码顺序。

[0134] 频谱系数是经无噪声编码,及始于最低频系数及前进至最高频系数而传输(例如,在位串流中)。

[0135] 得自进阶音频编码(例如,使用修正离散余弦变换获得,如 ISO/IEC14496-3,部分 3,子部分 4 讨论)的系数被储存在称作“x_ac_quant[g][win][sfb][bin]”的数组中,而无噪声编码码字组(例如 acod_m、acod_r)的传输顺序,使得当其是以接收且储存于该数组的顺序解码时,“bin”(频率指数)为最快递增指数,而“g”为最慢递增指数。

[0136] 与较低频相关联的频谱系数是比与较高频相关联的频谱系数更早编码。

[0137] 得自变换编码激励(tcx)的系数被直接储存于数组 x_tcx_invquant[win][bin],而无噪声编码码字组的传输顺序,使得当其是以接收且储存于该数组的顺序解码时,“bin”为最快递增指数,而“win”为最慢递增指数。换言之,若频谱值描述语音编码器的线性预测滤波器的变换编码激励,则频谱值 a 是与变换编码激励的相邻且递增的频率相关联。

[0138] 与较低频相关联的频谱系数是比与较高频相关联的频谱系数更早编码。

[0139] 值得注意,音频编码器 200 可被配置为应用由算术解码器 230 所提供的已解码的频域音频表示 232,用于使用频域至时域信号变换而“直接”产生时域音频信号表示,及用于使用频域至时域解码器及由频域至时域信号变换器的输出所激励的线性预测滤波器二者而“间接”提供音频信号表示。

[0140] 换言之,此处详细讨论其功能的算术解码器 200 极为适合用于解码以频域编码的音频内容的时频域表示的频谱值,及用于提供线性预测滤波器的一刺激信号的时频域表示,该滤波器是适用于解码以线性预测域编码的语音信号。如此,算术解码器是极为适合用于音频解码器,该音频解码器可处理频域编码音频内容及线性预测频域编码音频内容(变换编码激励线性预测域模式)。

[0141] 6.3. 依据图 5a 及图 5b 的上下文初始化

[0142] 后文中,将叙述在步骤 310 中执行的上下文初始化(也标示为“上下文映射”)。

[0143] 上下文初始化包含依据运算法则“arith_map_context ()”，在过去上下文与目前上下文之间的映射，显示于图 5a。如图可知，目前上下文储存于通用变量 q[2][n_context]，其是呈现具有 2 的第一维度及 n_context 的第二维度的数组。过去上下文储存于变量 qs[n_context]，其是呈现具有 n_context 维度的表形式。变量“previous_lg”描述过去上下文的频谱值数目。

[0144] 变量“lg”描述该帧内要解码的频谱系数数目。变量“previous_lg”描述前一帧的频谱行的先前数目。

[0145] 上下文的映射可依据运算法则“arith_map_context ()”进行。此处须注意，若与目前(例如，经频域编码的)音频帧相关联的频谱值数目是与对 i=0 至 i=lg-1 的前一个音频帧相关联的频谱值数目相等，则函数“arith_map_context ()”将目前上下文数组 q 的登录项目 q[0][i] 设定为过去上下文数组 qs 的值 qs[i]。

[0146] 然而，若目前音频帧相关联的频谱值数目是与前一个音频帧相关联的频谱值数目不等，则执行更复杂的映射。但此种情况下，有关映射细节与本发明的关键构想并非特别相关，故参考图 5a 的伪程序代码的细节。

[0147] 6.4. 依据图 5b 及图 5c 的状态值运算

[0148] 后文中，将更详细叙述状态值运算 312a。

[0149] 须注意，第一状态值 s (如图 3 所示) 可获得函数“arith_get_context (I,lg,arith_reset_flag,N/2)”作为返回值，其伪程序码表示是显示在图 5b 及图 5c 中。

[0150] 有关状态值的运算，也参考图 4，其显示用于状态评估的上下文。图 4 显示频谱值在时间及频率这二者上的二维表示。横坐标 410 描述时间，及纵坐标 412 描述频率。如图 4 可知，要解码的频谱值 420 是与时间指数 t0 及频率指数 i 相关联。如图可知，对时间指标 t0 而言，当具有频率指数 i 的频谱值 420 要被解码时，具有频率指数 i-1、i-2 及 i-3 的重元组已经解码。如由图 4 可知，在频谱值 420 被解码之前，具有时间指数 t0 及频率指数 i-1 的频谱值 430 已经解码，而频谱值 430 被考虑在用于频谱值 420 的解码的上下文。同理，在频谱值 420 被解码之前，具有时间指数 t0 及频率指数 i-2 的频谱值 434 已经解码，而频谱值 434 被考虑在用于频谱值 420 的解码的上下文。类似地，在频谱值 420 被解码之前，具有时间指数 t-1 及频率指数 i-2 的频谱值 440、具有时间指数 t-1 及频率指数 i-1 的频谱值 444、具有时间指数 t-1 及频率指数 i 的频谱值 448、具有时间指数 t-1 及频率指数 i+1 的频谱值 452、具有时间指数 t-1 及频率指数 i+2 的频谱值 456 已经解码，并且被考虑在用于频谱值 420 的解码的上下文的判定。当频谱值 420 解码时已经解码且被考虑用于上下文的频谱值(频谱是数)是以影线方形显示。相反地，(当频谱值 420 解码时)若干其它已经解码的频谱值是以具有虚线的方形显示；而(当频谱值 420 解码时)其它尚未解码的频谱值是以具有虚线的圆形显示，则并未用来判定用于解码频谱值 420 的上下文。

[0151] 但须注意虽言如此，若干这些尚未用于解码频谱值 420 的上下文的“常规”(或“正常”)运算的频谱值可被评估用于检测多个事先解码相邻频谱值其是单独地或共同地满足有关其幅度的预定状况。

[0152] 现在参考图 5b 及图 5c，该些图显示呈伪程序代码形式的函数“arith_get_context ()”的函数性，将叙述有关由函数“arith_get_context ()”执行的第一上下文值“s”的计算的进一步细节。

[0153] 须注意，函数“arith_get_context ()”接收要解码的频谱值的指数 i 作为输入变量。指数 i 典型地为频率指数。输入变量 lg 描述(针对一目前音频帧)预期量化系数的(总)数目。变数 N 描述变换的行数。标记“arith_reset_flag”指示该上下文是否应重置。函数“arith_get_context”提供表示连锁并置(concatenated)状态指数 s 及预测位平面位准 lev0 的变量“t”作为输出值。

[0154] 函数“arith_get_context ()”使用整数变量 a0、c0、c1、c2、c3、c4、c5、c6、lev0、及“region”。

[0155] 函数“arith_get_context ()”包含第一算术重置处理 510、一组多个事先解码相邻零频谱值的检测 512、第一变量设定 514、第二变量设定 516、位准调适 518、区值设定 520、位准调适 522、位准限制 524、算术重置处理 526、第三变量设定 528、第四变量设定 530、第五变量设定 532、位准调适 534 及选择返回值运算 536 作为主功能方块。

[0156] 在第一算术重置处理 510 中，检验是否设定算术重置标记“arith_reset_flag”，而要解码的频谱值的指标是等于零。此种情况下，返回零上下文值，及舍弃该功能。

[0157] 在一组多个事先解码零频谱值的检测 512，该功能唯有在算术重置标记为无效且要解码的频谱值指数 i 是非零时才执行，名为“flag”的变量被初始化为 1，如参考标号 512a 所示；及要被评估的频谱值一区经判定，如参考标号 512b 所示。随后，如参考标号 512b 所示而判定的该区频谱值是经评估，如参考标号 512c 所示。若发现有足够一区事先解码零频谱值，则返回 1 上下文值，如参考标号 512d 所示。举例而言，上频率指数边界“lim_max”设定为 i+6，除非要被解码的频谱值指数 i 是接近最大频率指数 lg-1，该种情况下，对上频率指数边界作特殊设定，如参考标号 512b 所示。此外，下频率指数边界“lim_min”设定为 -5，除非要解码的频谱值指数 i 是接近零(i+lim_min<0)，该种情况下，对下频率指数边界 lim_min 作特殊设定，如参考标号 512b 所示。当评估步骤 512b 所判定的该区频谱值时，首先对下频率指数边界 lim_min 与零之间的负频率指数 k 执行评估。对 lim_min 与零间的频率指数 k，证实上下文值 q[0][k].c 与 q[1][k].c 中的至少一个是否等于零。然而，若对 lim_min 与零间的任何频率指数 k，上下文值 q[0][k].c 与 q[1][k].c 二者皆非为零，则结论是并无足够的零频谱值组群，进而舍弃评估 512c。随后，评估零与 lim_max 间的频率指数的上下文值 q[0][k].c。若发现零与 lim_max 间的频率指数的任何上下文值 q[0][k].c 非零，则结论是并无足够的成组事先解码零频谱值，进而舍弃评估 512c。但若发现对 lim_min 与零间的每个频率指数 k，有至少一个上下文值 q[0][k].c 或 q[1][k].c 等于零，且若对零与 lim_max 间的每个频率指数 k 有零上下文值 q[0][k].c，则结论是有足够的成组事先解码零频谱值。据此，返回上下文值 1 来指示此种状况，而不再作任何额外计算。换言之，若识别有足够一组多个上下文值 q[0][k].c、q[1][k].c 具有零值，则跳过计算 514、516、518、520、522、524、526、528、530、532、534、536。换言之，回应于检测到满足预定状况，则与事先解码频谱值不相干地来判定描述上下文状态的所返回的上下文值。

[0158] 否则，即，若无足够成组上下文值 q[0][k].c、q[1][k].c 具有零值，则至少部分地执行运算 514、516、518、520、522、524、526、528、530、532、534、536。

[0159] 在第一变量设定 514，该步骤是若(且仅若)要被解码的频谱值指数 i 小于 1 才选择性执行，变量 a0 被初始化为上下文值 q[1][i-1]，及变量 c0 被初始化具有变量 a0 的绝对值。变量“lev0”被初始化为零值。随后，若变量 a0 包含较大的绝对值，即小于 -4，或大于

等于 4，则变量“lev0”及 c0 递增。变量“lev0”及 c0 的递增是迭代进行，直至变量 a0 通过朝右位移运算而进入 -4 至 3 的范围为止(步骤 514b)。

[0160] 随后，变量 c0 及“lev0”分别限于最大值 7 及 3 (步骤 514c)。

[0161] 若要被解码的频谱值的指数值 i 等于 1 并且算术重置标记(“arith_reset_flag”)有效，则返回上下文值，其是单纯基于变量 c0 及 lev0 运算(步骤 514d)。如此，只有具有与要解码的频谱值相同的时间指数及具有频率指数比要被解码的频谱值的频率指数 i 小 1 的单一事先解码频谱值被考虑用于上下文运算(步骤 514d)。否则，即，若无算术重置函数，则初始化变量 c4 (步骤 514e)。

[0162] 总结而言，在第一变量设定 514，变量 c0 及“lev0”是依事先解码频谱值初始化，解码用于与目前要被解码的频谱值相同帧，及用于前一个频谱仓 i-1。变量 c4 是依事先解码频谱值被初始化，解码用于前一个音频帧(具有时间指数 t-1)，及具有频率是低于(例如达一个频率仓)与目前要被解码的频谱值相关联的频率。

[0163] 若(且仅若)目前要被解码的频谱值的频率指数是大于 1，才选择性地执行的第二变量设定 516，包含变量 c1 及 c6 的初始化及变量 lev0 的更新。变量 c1 是依据目前音频帧的事先解码频谱值相关联的上下文值 q[1][i-2].c 更新，其频率是小于(例如，达 2 频率仓)目前要被解码的频谱值频率。类似地，变量 c6 是依据描述前一个帧(具有时间指标 t-1)的事先解码频谱值的上下文值 q[0][i-2].c 初始化，其相关频率是小于(例如达 2 频率仓)目前要被解码的频谱值频率。此外，位准变量“lev0”被设定为与目前帧的事先解码频谱值相关联的位准值 q[1][i-2].l，若 q[1][i-2].l 大于 lev0，则其相关频率是小于(例如达 2 频率仓)目前要被解码的频谱值频率。

[0164] 若(且仅若)要被解码的频谱值的指标 i 大于 2，位准调适 518 及区值设定 520 被选择性地执行。在位准调适 518，若与目前帧的事先解码频谱值相关联的位准值 q[1][i-3].l 大于位准值 lev0，则位准变数”lev0”是增至 q[1][i-3].l 值，其相关频率是小于(例如达 3 频率仓)目前要被解码的频谱值频率。

[0165] 在该区值设定 520，变量“区(region)”是依据评估设定，其中多个频谱区中的频谱区，布置目前要被解码的频谱值。举例而言，若发现目前要被解码的频谱值是与在该些频率仓的第一(最下)象限(0 ≦ i<N/4)频率仓(具有频率仓指数 i)相关联，则区变量“区”设定为零。否则，若目前要被解码的频谱值是与在该等频率仓的第二象限(N/4 ≦ i<N/2)频率仓相关联，则区变量设定为值 1。否则，若目前要被解码的频谱值是与在该些频率仓的第二(上半)半部(N/2 ≦ i<N)频率仓相关联，则区变量设定为 2。如此，区变量是依据目前要被解码的频谱值的频率区相关联的频率区的评估而设定。可区别两个以上频率区。

[0166] 若(且仅若)目前要被解码的频谱值包含大于 3 的指标，则执行额外位准调适 522。此种情况下，若位准值 q[1][i-4].l (其是与目前帧的事先解码频谱值相关联，而其是有关一种频率，该频率例如是比目前要被解码的频谱值相关联的频率小例如 4 频率仓)是大于目前位准“lev0”，则位准变量“lev0”增加(设定至值 q[1][i-4].l) (步骤 522)。位准变量“lev0”限于最大值 3 (步骤 524)。

[0167] 若检测到算术重置状况及目前要被解码的频谱值的指标 i 大于 1，则依据变量 c0、c1、lev0，以及依据区变量“区”而返回该状态值(步骤 526)。如此，若给定算术重置状况，则任何先前帧的事先解码频谱值不予考虑。

[0168] 在第三变量设定 528，变量 c2 设定为上下文值 q[0][i].c，其是与前一音频帧（具有时间指数 t-1）的事先解码频谱值相关联，该事先解码频谱值是与目前要被解码的频谱值的相同频率相关联。

[0169] 在第四变量设定 530，除非目前要被解码的频谱值是与最高可能频率指数 lg-1 相关联，否则变量 c3 设定为上下文值 q[0][i+1].c，其是与具有频率指数 i+1 的前一个音频帧的事先解码频谱值相关联。

[0170] 在第五变量设定 532，除非目前要被解码的频谱值的频率指数 i 是太过接近最大频率指数（即，具有频率指数值 lg-2 或 lg-1），否则变量 c5 设定为上下文值 q[0][i+2].c，其是与具有频率指数 i+2 的前一个音频帧的事先解码频谱值相关联。

[0171] 若频率指数 i 等于零（即，若目前要被解码的频谱值为最低频谱值），则进行位准变量“lev0”的额外调适。此种情况下，若变量 c2 或 c3 具有值 3（其指示与目前欲解码的频谱值相关联的频率比较时，与相同频率或甚至更高频率相关联的的前一音频帧的事先解码频谱值具有较大值），则位准变量“lev0”自零增至 1。

[0172] 在选择性返回值运算 536，返回值的运算是依据目前要被解码的频谱值的指标 i 是否具有值零、1、或更大值。若指标 i 具有零值，则返回值是依据变量 c2、c3、c5 及 lev0 运算，如参考标号 536a 所示。若指标 i 具有值 1，则返回值是依据变量 c0、c2、c3、c4、c5、及 lev0 运算，如参考标号 536b 所示。若指标 i 具有非零或非 1 的值，则返回值是依据变量 c0、c2、c3、c4、c1、c5、c6、“区”及 lev0 运算（参考标号 536c）。

[0173] 综上所述，上下文值运算“arith_get_context ()”包含一组多个事先解码零频谱值（或至少足够小的频谱值）的检测 512。若找到一组足够事先解码零频谱值，通过设定返回值为 1 而指示特殊上下文的存在。否则进行上下文值运算。通常，在上下文值运算中，指标值 i 被评估从而判定须评估多少个事先解码频谱值。举例言之，若目前要被解码的频谱值的频率指数 i 接近下边界（例如零）或接近上边界（例如 lg-1），则减少所评估的事先解码频谱值数目。此外，即便目前要被解码的频谱值的频率指数 i 足够远离最小值，则通过区值设定 520 区别不同的频谱区。据此，考虑不同频谱区（例如第一、低频率频谱区；第二、中频率频谱区；及第三、高频率频谱区）的不同统计性质。作为返回值的上下文值的计算是取决于变量“区”，使得该返回的上下文值是取决于该目前要被解码的频谱值是在第一预定频率区还是在第二预定频率区（或在任何其它预定频率区）。

[0174] 6.5. 映射规则选择

[0175] 后文中，将描述映射规则的选择，例如描述码值至符号码的映射的累积频率表。映射规则的选择是依据上下文状态进行，该上下文状态是以状态值 s 或 t 描述。

[0176] 6.5.1. 使用依据图 5d 的运算法则的映射规则选择

[0177] 后文中，将说明依据图 5d 使用函数“get_pk”选择映射规则。须注意，可执行函数“get_pk”从而在图 3 的运算法则的子运算法则 312ba 中获得值“pki”。如此，函数“get_pk”可取代图 3 的运算法则中的函数“arith_get_pk”。

[0178] 也须注意，依据图 5d 的函数“get_pk”可评估依据图 17（1）及图 17（2）的表“ari_s_hash[387]”及依据图 18 的表“ari_gs_hash”[225]。

[0179] 函数“get_pk”接收状态值 s 作为输入变量，该状态值 s 可通过依据图 3 的变量“t”与根据图 3 的变量“lev”、“lev0”组合而获得。函数“get_pk”也被配置为返回变量“pki”

值(其标示映射规则或累积频率表)作为返回值。函数“get_pk”被配置为将状态值 s 映射至映射规则指数值“pki”。

[0180] 函数“get_pk”包含第一表评估 540,及第二表评估 544。第一表评估 540 包含变量初始化 541,其中变量 i_min、i_max、及 i 被初始化,如参考标号 541 所示。第一表评估 540 也包含迭代表搜寻 542,在该过程判定是否存在匹配状态值 s 的表“ari_s_hash”的登录项目。若在迭代表搜寻 542 期间识别此种匹配,则舍弃函数 get_pk,其中通过匹配状态值 s 的表”ari_s_hash”的登录项目而判定该函数的返回值,稍后详述。然而,若在迭代表搜寻 542 期间并未找到状态值 s 与表”ari_s_hash”的登录项目间的完美匹配,则执行边界登录项目检查 543。

[0181] 现在转向第一表评估 540 的细节,可知由变量 i_min 及 i_max 界定搜寻区间。只要由变量 i_min 及 i_max 界定搜寻区间够大,则重复迭代表搜寻 542,若条件 i_max-i_min>1,则该状况为真。随后,至少约略近似地设定变量 i 来标示该区间的中点(i=i_min+(i_max-i_min)/2)。随后,设定变量 j 为由数组“ari_s_hash”位在变量 i 所标示的数组位置所判定的一值(参考标号 542)。此处须注意,表“ari_s_hash”的各个登录项目描述二者,即,与该表登录项目相关联的状态值,及与该表登录项目相关联的映射规则指数值。与该表登录项目相关联的状态值是由该表登录项目的最高有效位(位 8-31)描述;而映射规则指数值是由该表登录项目的较低位(例如位 0-7)描述。下边界 i_min 或上边界 i_max 是依据状态值 s 是否小于由该表“ari_s_hash”的变量 i 所参考的登录项目“ari_s_hash[i]”的最高有效 24 位所描述的状态值而调适。举例言之,若状态值 s 小于由登录项目“ari_s_hash[i]”的最高有效 24 位所描述的状态值,则该表区间的上边界 i_max 设定为值 i。如此,迭代表搜寻 542 的下次迭代的表区间被限于针对迭代表搜寻 542 的本次迭代所使用的表区间(自 i_min 至 i_max)的下半。相反地,若状态值 s 大于由表登录项目“ari_s_hash[i]”的最高有效 24 位所描述的状态值,则迭代表搜寻 542 的下次迭代的表区间的下边界 i_min 设定为值 i,使得目前表区间(在 i_min 至 i_max 间)的上半被用作针对下次迭代表搜寻的表区间。然而,若发现状态值 s 与由表登录项目“ari_s_hash[i]”的最高有效 24 位所描述的状态值相等,则由函数“get_pk”返回由表登录项目“ari_s_hash[i]”的最低有效 8 位所描述的映射规则指数值,进而舍弃该函数。

[0182] 迭代表搜寻 542 被重复,直至由变量 i_min 与 i_max 所界定的表区间足够小为止。

[0183] (可选地)执行边界登录项目检查 543 来补偿迭代表搜寻 542。若迭代表搜寻 542 完成后,指数变量 i 等于指数变量 i_max,则作最后检查状态值 s 是否等于由表登录项目“ari_s_hash[i_min]”的最高有效 24 位所描述的状态值,及此种情况下,返回由表登录项目“ari_s_hash[i_min]”的最低有效 8 位所描述的映射规则指数值作为函数“get_pk”的结果。相反地,若指数变量 i 与指数变量 i_max 不同,则执行检查状态值 s 是否等于由表登录项目“ari_s_hash[i_max]”的最高有效 24 位所描述的状态值,及此种情况下,返回由表登录项目“ari_s_hash[i_max]”的最低有效 8 位所描述的映射规则指数值作为函数“get_pk”的返回值。

[0184] 但须注意,边界登录项目检查 543 整体上可视为可选的。

[0185] 在第一表评估 540 之后,执行第二表评估 544,除非在第一表评估 540 期间出现“直接命中”,该种情况下,状态值 s 等于由表“ari_s_hash”的登录项目(或更明确地,由其 24

最高有效位）所描述的状态值中的一个。

[0186] 第二表评估 544 包含变量初始化 545，其中指数变量 i_min、i 及 i_max 被初始化，如参考标号 545 所示。第二表评估 544 也包含迭代表搜寻 546，在该过程中，搜寻表“ari_gs_hash”的一登录项目，该登录项目表示与状态值 s 相同的状态值。最后，第二表评估 544 包含返回值判定 547。

[0187] 只要由变量 i_min 及 i_max 界定的表区间够大（例如只要 i_max-i_min>1），则重复迭代表搜寻 546。在迭代表搜寻 546 的重复中，变量 i 设定为由 i_min 及 i_max 所界定的该表区间的中点（步骤 546a）。随后，表“ari_gs_hash”的变量 j 是位于指数变量 i 所判定的表位置获得（546b）。换言之，表登录项目“ari_gs_hash[i]”是位于由表指数 i_min 及 i_max 所界定的该目前表区间中点的一表登录项目。随后，判定针对迭代表搜寻 546 的下次迭代的表区间。为了达成此目的，若状态值 s 小于由表登录项目“j=ari_gs_hash[i]”的最高有效 24 位所描述的状态值，则描述该表区间的上边界的指数值 i_max 被设定为值 i（546c）。换言之，目前表区间的下半被选作针对迭代表搜寻 546 的下次迭代的新表区间（步骤 546c）。否则，若状态值 s 大于由表登录项目“j=ari_gs_hash[i]”的最高有效 24 位所描述的状态值，则指数值 i_min 被设定为值 i。如此，目前表区间的上半被选择作为针对迭代表搜寻 546 的下次迭代的新表区间（步骤 546d）。然而，若发现状态值 s 与由表登录项目“j=ari_gs_hash[i]”的最高有效 24 位所描述的状态值相等，则指数变量 i_max 设定为值 i+1 或设定为值 224（若 i+1 大于 224），且舍弃迭代表搜寻 546。然而，若状态值 s 与由“j=ari_gs_hash[i]”的 24 最高有效位所描述的状态值不同，则除非该表区间过小（i_max-i_min ≦ 1），否则迭代表搜寻 546 是以由已更新的指标值 i_min 及 i_max 所界定的新设定表区间重复。如此，表区间（由 i_min 及 i_max 所界定）的区间大小迭代地缩小直至检测到“直接命中”（s== (j>>8)），或直至区间达到最小容许大小（i_max-i_min ≦ 1）为止。最后，在舍弃了迭代表搜寻 546 后，判定表登录项目“j=ari_gs_hash[i_max]”，及由该表登录项目“j=ari_gs_hash[i_max]”的 8 个最低有效位所描述的映射规则指数值被返回作为函数“get_pk”的返回值。如此，映射规则指数值是依据在迭代表搜寻 546 完成或舍弃后，表区间（由 i_min 及 i_max 所界定）的上边界 i_max 判定。

[0188] 都使用迭代表搜寻 542、546 的前述表评估 540、544 允许以极高的运算效率检验表“ari_s_hash”及“ari_gs_hash”是否存在一给定的有效状态。更明确地，即便于最恶劣情况下，表存取运算次数仍可维持合理地小。已发现表“ari_s_hash”及“ari_gs_hash”的数值定序，允许加速搜寻适当哈希值。此外，表的大小可维持较小，原因在于不需要在表“ari_s_hash”及“ari_gs_hash”中包括逸出符号。如此，即便有大量不同状态，仍可建立有效上下文哈希机制：在第一阶段（第一表评估 540），进行针对直接命中的搜寻（s== (j>>8)）。

[0189] 在第二阶段（第二表评估 544），状态值 s 的范围可映射至映射规则指数值。如此，可执行表“ari_s_hash”中有相关联的登录项目的特别有效状态、与基于范围的处理的较低有效状态的良好平衡处置。据此，函数“get_pk”组成映射规则选择的有效实现。

[0190] 有关任何进一步细节，请参考图 5d 的伪程序代码，其是以依据众所周知程序语言 C 的表示而表示函数“get_pk”的函数性。

[0191] 6.5.2. 使用依据图 5e 的运算法则的映射规则选择

[0192] 后文中，将参考图 5e 叙述映射规则选择的另一项运算法则。须注意，依据图 5e 的

运算法则“arith_get_pk”接收描述上下文状态的一状态值 s 作为输入变量。函数“arith_get_pk”提供概率模型的指数“pki”作为输出值或返回值，该指数可为用以选择映射规则的指数(例如累积频率表)。

[0193] 须注意，依据图 5e 的函数“arith_get_pk”可具有图 3 函数“value_decode”的函数“arith_get_pk”的函数性。

[0194] 也须注意，函数“arith_get_pk”例如可评估依据图 20 的表 ari_s_hash 及依据图 18 的表 ari_gs_hash。

[0195] 依据图 5e 的函数“arith_get_pk”｜包含第一表评估 550 及第二表评估 560。在第一表评估 550，对于表 ari_s_hash 作线性扫描，获得该表的登录项目 j=ari_gs_hash[i]。若由表 ari_s_hash 的一表登录项目 j=ari_gs_hash[i] 的最高有效 24 位描述的状态值等于状态值 s，则返回由该所识别的表登录项目 j=ari_gs_hash[i] 的最低有效 8 位所描述的映射规则指数值“pki”，及舍弃函数“arith_get_pk”。据此，除非识别“直接命中”(状态值 s 等于表登录项目 j 的最高有效 24 位描述的状态值)，否则表 ari_s_hash 的全部 387 登录项目是以上升顺序评估。

[0196] 若在第一表评估 550 未识别直接命中，则执行第二表评估 560。在第二表评估过程中，执行线性扫描，登录项目指数 i 自零线性递增至 224 最大值。在第二表评估期间，读取针对表 i 的表“ari_gs_hash”的登录项目“ari_gs_hash[i]”，且评估表登录项目“j=ari_gs_hash[i]”，其中判定由表登录项目 j 的 24 最高有效位所表示的状态值是否大于状态值 s。若属此种状况，则返回由表登录项目 j 的 8 最低有效位所描述的映射规则指数值作为函数“arith_get_pk”的返回值，及舍弃函数“arith_get_pk”的执行。然而，若状态值 s 不小于由目前表登录项目 j=ari_gs_hash[i] 的 24 最高有效位所描述的状态值，则通过递增表指数 i 而继续扫描对于表 ari_gs_hash 的登录项目。然而，若状态值 s 大于或等于由表登录项目 ari_gs_hash 所描述的任一个状态值，则返回由表 ari_gs_hash 的 8 最低有效位所界定的映射规则指数值“pki”作为函数“arith_get_pk”的返回值。

[0197] 总而言之，依据图 5e 的函数“arith_get_pk”执行二步骤式哈希。在第一步骤，执行针对直接命中的搜寻，其中判定状态值 s 是否等于由第一表“ari_gs_hash”的任一登录项目所描述的状态值。若在第一表评估 550 中识别直接命中，则自第一表“ari_s_hash”获得返回值，而舍弃函数“arith_get_pk”。然而，若在第一表评估 550 未识别直接命中，则执行第二表评估 560。在第二表评估，执行基于范围的评估。第二表“ari_gs_hash”的接续登录项目界定范围。若发现状态值 s 是落入此一范围(其是由下述事实指示，由目前表登录项目“j=ari_gs_hash[i]”的 24 最高有效位所描述的状态值大于状态值 s)，则送返由表登录项目“j=ari_gs_hash[i]”的 8 最低有效位所描述的映射规则指数值“pki”。

[0198] 6.5.3. 使用依据图 5f 的运算法则的映射规则选择

[0199] 依据图 5f 的函数“get_pk”实质上相当于依据图 5e 的函数“arith_get_pk”。因而，参考前文讨论。有关进一步细节，请参考图 5f 的伪程序表示。

[0200] 须注意，依据图 5f 的函数“get_pk”可替代图 3 称作为函数“value_decode”的函数“arith_get_pk”。

[0201] 6.6. 依据图 5g 的函数“arith_decode ()”

[0202] 后文中，将参考图 5g 讨论函数“arith_decode ()”的函数性的进一步细节。须了

解，函数“arith_decode（）”使用助手函数“arith_first_symbol（void）”，若为该序列中的第一符号则返回 TRUE，否则返回 FALSE。函数“arith_decode （）”也使用助手函数“arith_get_next_bit （void）”，其获取且提供该位串流的下一位。

[0203] 此外，函数“arith_decode （）”使用全局变量“low”、“high”及“value”。此外，函数“arith_decode （）”接收变量“cum_freq[]”作为输入变量，其指向所选累积频率表的第一登录项目或元素（具有元素指数或登录项目指数 0）。同样，函数“arith_decode （）”使用输入变量“cfl”，其指示以变量“cum_freq[]”标示的所选累积频率表的长度。

[0204] 函数“arith_decode（）”包含变量初始化 570a 作为第一步骤，若助手函数“arith_first_symbol （）”指示一序列符号的第一符号是经解码，则执行该步骤。变量初始化 550a 依据多个例如 20 位而初始化变量“value”，该些位是使用助手函数“arith_get_next_bit”而得自位串流，使得该变量“value”具有该些位所表示的值。同样，变量“low”被初始化为具有 0 值，而变量“high”被初始化为具有 1048575 值。

[0205] 在第二步骤 570b，变量“range”设定为比变量｜“high”与”low”数值间的差值大 1 的值。变量“cum”设定为一值，其表示变量“low”值与变量“high”值间的变量“value”值的相对位置。如此，变量“cum”例如依据变量“value”值而具有 0 至 2^{16} 间的一值。

[0206] 指标器 p 被初始化为一值，该值是比所选累积频率表的起始地址小 1。

[0207] 运算法则“arith_decode （）”也包含迭代累积频率表搜寻 570c。该迭代累积频率表搜寻被重复，直至变量 cfl 小于或等于 1 为止。在迭代累积频率表搜寻 570c，指数器变量 q 设定为一值，该值等于指数器变量 p 的目前值与变量“cfl”值的一半的和数。若所选累积频率表的由指数器变量 q 所寻址的该登录项目 *q 的值大于变量“cum”的值，则指数器变量 p 被设定至指数器变量 q 的值，而变量“cfl”递增。最后，变量“cfl”向右位移一位，由此有效地将变量“cfl”除以 2，及忽略取模（modulo）部分。

[0208] 如此，迭代累积频率表搜寻 570c 有效地比较变量“cfl”值与该所选累积频率表的多个登录项目，从而识别出该所选累积频率表内部是由该累积频率表的登录项目所画界的一区间，使得值 cum 位在所识别的区间内。如此，该所选累积频率表的登录项目界定区间，其中个别符号值是与该所选累积频率表的各个区间相关联。同样，该累积频率表的两相邻值之间的区间宽度界定与该区间相关联的符号的概率，使得所选累积频率表全体界定不同符号（或符号值）的概率分布。有关可用累积频率表的细节将参考图 19 讨论如下。

[0209] 再次参考图 5g，符号值是从指数器变量 p 导出，其中该符号值的导算是如参考标号 570d 所示。如此，指数器变量 p 值与起始地址“cum_freq”间的差值被评估，从而获得该符号值，其以变量“symbol”表示。

[0210] 运算法则“arith_decode”也包含变量“high”及“low”的调适 570e。若由变量“symbol”表示的符号值非 0，则变量“high”被更新，如参考标号 570e 所示。变量“high”被设定为由变量“low”、变量“range”及所选累积频率表的具有指数“symbol-1”的登录项目的值所判定的一值。变量“low”增加，其中增加幅度是由变量“range”及所选累积频率表的具有指数“symbol”的登录项目判定。如此，变量“low”与“high”的值间的差值是依据所选累积频率表的两相邻登录项目的数值差而调整。

[0211] 据此，若检测到具有低概率的一符号值，则变量“low”与“high”的值之间的区间缩小至狭窄宽度。相反地，若检测到的符号值包含相对大的概率，则变量“low”与“high”

的值之间的区间宽度设定为相对较大值。再度，变数“low”与“high”的值之间的区间宽度是取决于检测到的符号及相对应的累积频率表的登录项目。

[0212] 运算法则“arith_decode”也包含区间再标准化 570f，其中在步骤 570e 中测定的区间被迭代地位移及定标直至达到“断裂(break)”状况为止。在区间再标准化 570f，执行选择性向下位移运算 570fa。若变量“high”小于 524286 则不作为，而以区间大小增加运算 570fb 继续区间再标准化。然而，若变量“high”不小于 524286，并且变数“low”大于或等于 524286，则变数“values”、“low”、及“high”全部减 524286，使得由变量“low”及“high”所界定的区间向下位移，且使得变量“value”的值也向下位移。然而，若发现变量“high”不小于 524286，且变量“low”不大于或等于 524286，且变量“low”大于或等于 262143，且变量“high”小于 786429，则变数“value”、“low”、及“high”全部减 262143，使得由变量“low”及“high”所界定的区间向下位移，且使得变量“value”的值也向下位移。然而，若未满足前述任一种情况，则舍弃区间再标准化。

[0213] 然而，若满足步骤 570fa 评估的任一个前述件，则执行区间增加运算 570fb。在区间增加运算 570fb，变量“low”的值加倍。同样，变量“high”的值加倍，加倍结果递增 1。同样，变量“value”的值加倍(朝左位移 1 位)，及由助手函数“arith_get_next_bit”所得的位串流的一位被用作最低有效位。据此，变量“low”及“high”之间的区间大小被近似地加倍，及变量“value”的精度通过使用该位串流的一新位而增高。如前述，步骤 570fa 及 570fb 重复直至达“断裂”状况，即，直至变量“low”与“high”数值间的区间够大为止。

[0214] 有关运算法则“arith_decode ()”的函数性，须注意在步骤 570e，依据由变量“cum_freq”所参照的累积频率表的两相邻登录项目，变量“low”与“high”数值间的区间缩小。若所选累积频率表的两相邻值间的区间小，即，若相邻值较为靠近，则步骤 570e 所获得的变量“low”与“high”数值间的区间将相对较小。相反地，若累积频率表的两相邻登录项目较为远离，则步骤 570e 所获得的变量“low”与“high”数值间的区间将相对较大。

[0215] 结果，若步骤 570e 所获得的变量“low”与“high”数值间的区间为相对较小，则将执行大量的区间再标准化步骤来复位标该区间至“足够”的大小(使得不满足状况评估 570fa 的任一种状况)。如此，相对较大量的来自位串流的位将用来提高变量“value”的精度。相反地，若步骤 570e 所获得的区间大小为相对较大，则只需少量重复区间标准化步骤 570fa 及 570fb 从而将变量“low”与“high”数值间的区间再标准化为“足够”大小。如此，只有相对较少数量得来自位串流的位将用来提高变量“value”的精度，及准备下一个符号的解码。

[0216] 综上所述，若解码一符号(其包含较高概率且所选累积频率表的登录项目是与其大区间相关联)，则将只从位串流读取较少数量的位，来允许随后符号的解码。相反地，若解码一符号(其包含较低概率且所选累积频率表的登录项目是与其小区间相关联)，则将从位串流取得较大量的位来准备下一符号的解码。

[0217] 如此，累积频率表的登录项目反映不同符号的概率，同时也反映解码一序列符号所需位数目。通过依据上下文，亦即依据事先解码符号(或频谱值)而变量累积频率表，例如通过依据上下文而选择不同的累积频率表，可探讨不同符号间的随机相依性，其允许随后的(或相邻的)符号的特定位率有效编码。

[0218] 综上所述，已经参考图 5g 描述的函数“arith_decode ()”是连同累积频率表

“arith_cf_m[pki][]”调用，对应于由函数“arith_get_pk（）”所返回的指数“pki”，判定最高有效位平面值 m（其可被设定为由返回变量 |symbol”表示的符号值）。

[0219] 6.7. 逸出机制

[0220] 虽然已解码的最高有效位平面值 m（其可由函数“arith_decode（）”返回作为符号值）为逸出符号“ARITH_ESCAPE”，但解码另一个最高有效位平面值 m，及变量“lev”递增 1。据此，获得有关最高有效位平面值 m 的数值重要性及要被解码的较低有效位平面数目的信息。

[0221] 若逸出符号“ARITH_ESCAPE”经解码，则位准变量“lev”递增 1。如此，输入至函数“arith_get_pk”的状态值也经修正，由最高位（位 24 及以上）所表示的值对运算法则 312ba 的下次迭代增加。

[0222] 6.8. 依据图 5h 的上下文更新

[0223] 一旦频谱值完全被解码（即，全部最低有效位平面皆已经相加，上下文表 q 及 qs 是通过调用函数“arith_update_context（a,i,lg）”而更新）。后文中，将参考图 5h 描述有关函数“arith_update_context（a,i,lg）”的细节，其显示该函数的伪程序码表示。

[0224] 函数“arith_update_context（a,i,lg）”接收已解码的量化频谱系数 a、要被解码的频谱值（或已解码的频谱值）指数 i、及与目前音频帧相关联的频谱值（或频谱系数）的数目 lg 作为输入变量。

[0225] 在步骤 580，目前已解码的量化频谱值（或系数）a 被拷贝至上下文表或上下文数组 q。如此，上下文表 q 的登录项目 q[l][i] 设定为 a。同样，变量“a0”被设定为值“a”。

[0226] 在步骤 582，判定上下文表 q 的位准值 q[l][i].l。经由默认，将上下文表 q 的位准值 q[l][i].l 设定为零。然而，若目前已解码的频谱值 a 的绝对值大于 4，则位准值 q[l][i].l 递增。随着各次递增，变量“a”向右位移一位。重复位准值 q[l][i].l 的递增，直至变量 a0 的绝对值小于或等于 4 为止。

[0227] 在步骤 584，设定上下文表 q 的 2- 位上下文值 q[l][i].c。若目前已解码的频谱值 a 等于零，则 2- 位上下文值 q[l][i].c 被设定为零值。否则，若已解码的频谱值 a 的绝对值小于或等于 1，则 2- 位上下文值 q[l][i].c 设定为 1。否则，若目前已解码的频谱值 a 的绝对值小于或等于 3，则 2- 位上下文值 q[l][i].c 设定为 2。否则，即，若目前已解码的频谱值 a 的绝对值大于 3，则 2- 位上下文值 q[l][i].c 设定为 3。如此，2- 位上下文值 q[l][i].c 是通过目前已解码的频谱值 a 的极为粗糙的量化而获得。

[0228] 在接续步骤 586，此步骤仅在目前已解码的频谱值的指数 i 等于帧的系数（频谱值）数目 lg 时才执行，换言之，若帧的最末频谱值已经解码及核心模是线性预测域核心模（其是以“core_mode==1”指示），则登录项目 q[l][j].c 被拷贝入上下文表 qs[k]。参考标号 586 所示，执行拷贝，使得目前帧的频谱值数目 lg 被列入考虑用以将登录项目 q[l][j].c 拷贝至上下文表 qs[k]。此外，变量“previous_lg”具有值 1024。

[0229] 然而，可替换地，若目前已解码的频谱系数的指数 i 达 lg 值，及核心模是频域核心模（其是以“core_mode==1”指示），则上下文表 q 的登录项目 q[l][j].c 被拷贝入上下文表 qs[j]。

[0230] 此种情况下，变量“previous_lg”被设定为值 1024 与帧内频谱值数目 lg 间的最小值。

[0231] 6.9. 解码程序的摘要

[0232] 后文中，将简单摘述解码程序。有关其细节，请参考前文讨论及图 3、图 4 及图 5a 至图 5i。

[0233] 始于最低频率系数并且前进至最高频率系数，量化频谱系数 a 是无噪声式编码及传输。

[0234] 得自进阶音频编码（AAC）的系数储存于数组“x_ac_quant[g][win][sfb][bin]”，及无噪声编码码字组的传输顺序为当其是以所接收的及储存于数组的顺序解码时，bin 为最快递增指数，而 g 为最慢递增指数。指数 bin 表示频率仓。指数“sfb”表示定标因子带。指数“win”指示窗。指数“g”指示音频帧。

[0235] 得自变换编码激励的系数被直接储存在数组“x_tcx_invquant[win][bin]”，并且无噪声编码码字组的传输顺序为当其是以所接收的及储存于数组的顺序解码时，“bin”为最快递增指数，而“win”为最慢递增指数。

[0236] 首先，在上下文表或数组“qs”中所储存的过去上下文与目前帧 q 上下文（储存在上下文表或数组 q）间进行映射。过去上下文“qs”被储存在每一频率行（或每一频率仓）2- 位。

[0237] 在上下文表“qs”中所储存的过去上下文与储存在上下文表“q”的目前帧上下文间的映射是使用函数“arith_map_context（）”执行，其伪程序码表示是显示于图 5a。

[0238] 无噪声解码器输出带符号的量化频谱系数“a”。

[0239] 首先，基于环绕要解码的量化频谱系数的事先解码频谱系数，计算上下文状态。上下文状态 s 与由函数“arith_get_context（）”所返回值的首 24 位相对应。超过返回值的第 24 位的位与预测位平面位准 lev0 相对应。变量“lev”被初始化为 lev0。函数“arith_get_context”的伪程序码表示在图 5b 及图 5c 中示出。

[0240] 一旦状态 s 及预测位准“lev0”为已知，则使用函数“arith_decode（）”解码最高有效逐 2- 位平面值 m，被馈以与上下文状态相对应的概率模型相对应的适当累积频率表。

[0241] 对应关系是以函数“arith_get_pk（）”作出。

[0242] 函数“arith_get_pk（）”的伪程序码表示在图 5e 中示出。

[0243] 可替代函数“arith_get_pk（）”的另一个函数“get_pk”的伪程序码表示显示于图 5f。可替代函数“arith_get_pk（）”的另一个函数“get_pk”的伪程序码表示显示于图 5d。

[0244] 使用连同累积频率表“arith_cf_m[pki][]”被调用的函数“arith_decode（）”来解码值 m，此处“pki”是对应于由函数“arith_get_pk（）”（或另外，由函数“get_pk（）”）所返回的指数。

[0245] 算术编码器为使用以定标标度产生标签的方法的整数实现方式（例如参考 K.Sayood“Introduction to Data Compression”第三版 2006 年，ElsevierInc.）。图 5g 所示伪 C 码描述所使用的运算法则。

[0246] 当解码值 m 为逸出符号“ARITH_ESCAPE”时，解码另一个值 m，及变量“lev”递增 1。一旦值 m 并非逸出符号“ARITH_ESCAPE”，通过调用函数“arith_decode（）”连同累积频率表“arith_cf_r[]”达“lev”次，则其余位平面从最高有效位准至最低有效位准解码。例如，该累积频率表“arith_cf_r[]”可描述均衡概率分布。

[0247] 解码位平面 r 允许以下述方式改进事先解码值 m：

[0248]

```
a = m;
for (i=0; i<lev;i++) {
        r = arith_decode (arith_cf_r,2);
        a = (a<<1) | (r&1);
}
```

[0249] 一旦频谱量化系数已完全解码，则上下文表 q 或所储存的上下文 qs 由函数“arith_update_context ()”更新，用于下一个要解码的量化频谱系数。

[0250] 函数“arith_update_context ()”的伪程序码表示显示于图 5h。

[0251] 此外，定义的图例显示于图 5i。

[0252] 7. 映射表

[0253] 在依据本发明的实施例，特别优异的表“arith_s_hash”及“arith_gs_hash”及“ari_cf_m”是用于函数“get_pk”的执行，其已经参考图 5d 讨论；或用于函数“arith_get_pk”的执行，其已经参考图 5e 讨论；或用于函数“get_pk”的执行，其已经参考图 5f 讨论；或用于函数“arith_decode”的执行，其已经参考图 5g 讨论。

[0254] 7.1. 依据图 17 的表“arith_s_hash[387]”

[0255] 在图 17 的表中示出了表“arith_s_hash”特别优异的实现方式的内容，该表是用于已经参考图 5d 讨论的函数“get_pk”。须注意，图 17 的表列举表“arith_s_hash[387]”的 387 登录项目。也须注意，图 17 的表表示显示依元素指数排序的元素，使得第一值“0x00000200”是对应于具有元素指数（或表指数）0 的表登录项目“ari_s_hash[0]”，使得最末值“0x03D0713D”对应于具有元素指数或表指数 386 的表“ari_s_hash[386]”。进一步须注意，“0x”指示表“ari_s_hash”的表登录项目是以十六进制格式表示。此外，依据图 17 的表“ari_s_hash”的表登录项目是以数值顺序排列，从而允许执行函数“get_pk”的第一表评估 540。

[0256] 进一步须注意，表“ari_s_hash”的表登录项目的最高有效 24 位表示状态值，而最低有效 8 位表示映射规则指数值 pki。

[0257] 如此，表“ari_s_hash”的表登录项目描述一状态值“直接命中”映射至一映射规则指数值“pki”。

[0258] 7.2. 依据图 18 的表“ari_gs_hash”

[0259] 表“ari_gs_hash”的特佳实施例的内容显示于图 18 的表。此处须注意，表 18 的表列举表“ari_gs_hash”的登录项目。该些登录项目是由一维整数型登录项目指数（也标示为“元素指数”或“数组指表”或“表指标”）参照，例如标示以“i”。须注意，共含 225 登录项目的表“ari_gs_hash”极为适合用于由图 5d 所述的函数“get_pk”的第二表评估 544 使用。

[0260] 须注意，表“ari_gs_hash”的登录项目是以对零至 224 间的表指数值 i 的表指数 i 的上升顺序列举。项“0x”指示表登录项目是以十六进制格式描述。如此，第一表登录项目“0X000000401”对应于具有表指数 0 的表登录项目“ari_gs_hash[0]”，而最末表登录项目“0Xfffff3f”对应于具有表指数 224 的表登录项目“ari_gs_hash[224]”。

[0261] 也须注意，表登录项目是以数值型上升方式排序，使得表登录项目极为适合用于函数“get_pk”的第二表评估 544。表“ari_gs_hash”的表登录项目的最高有效 24 位描述状态值范围间的边界，而登录项目的 8 最低有效位描述与 24 最高有效位所界定的状态值范围相关联的映射规则指数值“pki”。

[0262] 7.3. 依据图 19 的表“ari_cf_m”

[0263] 图 19 显示一集合 64 个累积频率表“ari_cf_m[pki][9]”，其中一个是由音频编码器 100、700 或音频解码器 200、800 选用来执行函数“arith_decode”，即，用于最高有效位平面值的解码。图 19 所示的 64 累积频率表中的选定者利用表“cum_freq[]”的函数执行函数“arith_decode ()”。

[0264] 如自图 19 可知，各行表示有 9 个登录项目的累积频率表。举例言之，第一行 1910 表示针对“pki=0”的一累积频率表的 9 登录项目。第二行 1912 表示针对“pki=1”的一累积频率表的 9 登录项目。最后，第 64 行 1964 表示针对“pki=63”的一累积频率表的 9 登录项目。如此，图 19 有效表示针对“pki=0”至“pki=63”的 64 个不同累积频率表，其中 64 个累积频率表各自是以单行表示，及其中该些累积频率表各自包含 9 登录项目。

[0265] 在一行内部(例如，行 1910 或行 1912 或行 1964)，最左值描述累积频率表的第一登录项目，而最右值描述累积频率表的最末登录项目。

[0266] 如此，图 19 的表表示的各行 1910、1912、1964 表示由依据图 5g 的函数“arith_decode”使用的一累积频率表的登录项目。函数“arith_decode”的输入变量“cum_freq[]”描述表“ari_cf_m”的 64 累积频率表(9 登录项目的各行表示)中的哪个应当用于目前频谱系数的解码。

[0267] 7.4. 依据图 20 的表“ari_s_hash”

[0268] 图 20 显示表“ari_s_hash”的另一替代实例，其可组合依据图 5e 或图 5f 的替代函数“arith_get_pk ()”或“get_pk ()”使用。

[0269] 依据图 20 的表“ari_s_hash”包含 386 登录项目，其是以表指数的上升顺序列举于图 20。如此，第一表值“0x0090D52E”对应于具有表指数 0 的表登录项目“ari_s_hash[0]”，而最末表值“0x03D0513C”对应于具有表指数 386 的表登录项目“ari_s_hash[386]”。

[0270] “0x”指示表登录项目是以十六进制格式表示。表“ari_s_hash”的表登录项目的最高有效 24 位表示重要状态，而表“ari_s_hash”的登录项目最低有效 8 位表示映射规则指数值。

[0271] 据此，表“ari_s_hash”的登录项目描述重要状态映射至映射规则指数值“pki”。

[0272] 8. 性能评估及优点

[0273] 依据本发明的实施例使用如前文讨论的更新函数(或运算法则)及更新的表集合来获得运算复杂度、内存需求、与编码效率间的改良式折衷。

[0274] 概略言之，依据本发明的实施例形成一种改良式频谱无噪声编码。

[0275] 本说明书描述 CE 用于频谱系数的改良式频谱无噪声编码的实施例。所提示的方案是基于“原先”上下文式算术编码方案，如描述于 USAC 草拟标准工作草案 4，但显著减低内存需求(RAM、ROM)，同时维持无噪声编码效能。WD3(即，音频编码器的输出信号提供 USAC 草拟标准工作草案的位串流)的无损耗转码证实为可能。此处所述方案大致上可定标，允许内存需求与编码效能间的进一步替代折衷。依据本发明的实施例是针对替代如用于 USAC

草拟标准工作草案 4 的无噪声编码方案。

[0276] 此处所述算术编码方案是基于 USAC 草拟标准工作草案 4(WD4)的参考模型 0(RM0)中的方案。频谱系数先前于频率模型或时间模型为上下文。此一上下文用于算术编解码器(编码器或解码器)的累积频率表的选择。比较依据 WD4 的实施例,上下文模型化进一步改良,而保有符号概率的表重新训练。不同概率模型的数目自 32 增至 64。

[0277] 依据本发明的实施例将表大小(数据 ROM 需求)缩减至 900 个长度 32- 位字组或 3600 字节。相反地,依据 USAC 草拟标准的 WD4 实施例要求 16894.5 字组或 76578 字节。依据本发明的若干实施例,每个核心编码器信道的静态 RAM 需求自 666 字组(2664 字节)减至 72 字组(288 字节)。同时,可全然保有编码性能,与共 9 个运算点的总数据率相比,甚至可达约 1.04% 至 1.39% 的增益。全部工作草案 3 (WD3)位串流可以无损耗方式转码而不影响位储存限制。

[0278] 依据本发明的实施例所提示的方案可扩增 :内存需求与编码效能间的弹性折衷是可能的。通过加大表的大小从而进一步增加编码增益。

[0279] 后文中,将提供 USAC 草拟标准 WD4 的编码构想的简短讨论来协助了解此处所述构想的优点。在 USAC WD4 中,基于上下文的算术编码方案是用于量化频谱系数的无噪声编码。作为上下文,使用频率及时间上为先前的已解码的频谱系数。依据 WD4,最大数目 16 频谱系数被用作上下文,其中 12 个的时间在先。用于上下文的及要被解码的频谱系数二者是分组成 4- 重元组(即,四个频谱系数的频率相邻,参考图 10a)。上下文缩减且映射至一累积频率表,其然后用来解码频谱系数的下一个 4- 重元组。

[0280] 针对完整的 WD4 无噪声编码方案,需要 16894.5 字组(67578 字节)的内存需求(ROM)。此外,每个核心编码器信道的要求 666 字组(2664 字节)的静态 ROM 来储存下一帧状态。

[0281] 图 11a 的表表示描述用于 USAC WD4 算术编码方案的表。

[0282] 完整 USAC WD4 解码器的总内存需求估算为对不含程序代码的数据 ROM 为 37000 字组(148000 字节),而对静态 RAM 为 10000 至 17000 字组。显然,无噪声编码器表消耗总数据 ROM 需求的约 45%。该最大的个别表已经耗掉 4096 字组(16384 字节)。

[0283] 发现全部表组合的大小及最大的个别表二者皆超过由固定点芯片对低预算的可携式装置所提供的典型缓冲大小,其是在 8 至 32 千字节的典型范围(例如 ARM9e、TIC64xx 等)。这意味着表的集合可能并未储存在最快数据 RAM (其允许快速随机存取数据)。如此造成整个解码过程变慢。

[0284] 后文中,将简短叙述所提出的新颖方案。

[0285] 为了克服前述问题,提示一种改良式无噪声编码方案来替代 USAC 草拟标准 WD4 的方案。至于基于上下文的算术编码方案,其是基于 USAC 草拟标准 WD4 方案,但具有改良式方案特征用来自该上下文导出累积频率表。此外,上下文导算及符号编码是对单一频谱系数的粒度(granularity)执行(与如 USAC 草拟标准 WD4 所使用的 4- 重元组相反)。总计 7 个频谱系数用于上下文(至少在某些情况下)。通过减少映射关系,选出总计 64 概率模型或累积频率表(在 WD4 :32)中的一个。

[0286] 图 10b 显示用于所提出的方案,用于状态计算的上下文的图解代表图(其中,用于零区检测的上下文未显示于图 10b)。

[0287] 后文中，将简短说明有关内存需求缩减的讨论，该目的可使用所提出的编码方案达成。所提出的新方案具有总计 900 字组（3600 字节）的 ROM 需求（参考图 11b 的表，其描述用于所提出的编码方案的表）。

[0288] 与 USAC 草拟标准 WD4 的无噪声编码方案的 ROM 需求相比较，ROM 需求减少 15994.5 字组（64978 字节）（也参考图 12a，该图显示 USAC 草拟标准 WD4 的无噪声编码方案的 ROM 需求以及所提出的无噪声编码方案的 ROM 需求的图解代表图）。如此将完整 USAC 解码器的总 ROM 需求从约 37000 字组减少至约 21000 字组，或减少多于 43%（参考图 12b，其显示依据 USAC 草拟标准 WD4，以及依据本提案的总 USAC 解码器数据 ROM 需求的图解代表图）。

[0289] 此外，也减少下一帧（静态 RAM）的上下文导算所需信息量。依据 WD4，典型具有 16- 位分辨率的系数的完整集合（至多 1152）加至须要储存的每个 10- 位分辨率 4- 重元组的一组指数，其相加达到每个核心编码器信道（完整 USAC WD4 解码器 ：约 10000 至 17000 字组）666 字组（2664 字节）。

[0290] 用于依据本发明的实施例的新颖方案将持久信息减少至每个频谱系数只有 2- 位，其相加达到每个核心编码器信道总计 72 字组（288 字节）。对静态内存的需求可减少 594 字组（2376 字节）。

[0291] 后文中，将描述有关细码效率可能增高的若干细节。依据新颖提案的实施例的编码效率是对依据 USAC 草拟标准 WD3 的参考质量位串流作比较。该比较是基于参考软件解码器，利用转码器执行。有关依据 USAC 草拟标准 WD3 的无噪声编码与本案所提出的编码方案的比较细节，参考图 9，该图显示测试配置的示意代表图。

[0292] 虽然依据本发明的实施例相比于依据 USAC 草拟标准 WD3 或 WD4 的实施例，内存需求大减，但不仅维持编码效率，反而编码效率略增。编码效率平均增高 1.04% 至 1.39%。有关其细节请参考图 13a 的表，其显示依据本发明的实施例，使用工作草案算术编码器及音频编码器（例如 USAC 音频编码器），由 USAC 编码器所产生的平均位率的表表示。

[0293] 通过测量位储存的填补位准，显示所提出的无噪声编码可对每个运算点，无损耗地转码 WD3 位串流。有关其细节，参考图 13b 的表，其显示依据本发明的实施例的音频编码器及依据 USAC WD3 的音频编码器，位储存控制的表表示。

[0294] 每个运算模的平均位率的相关细节，以帧为基准的最小、最大及平均位率，及基于帧基准的最佳 / 最恶劣情况性能可参考图 14、15 及 16 的表，其中图 14 的表显示依据本发明的实施例的音频编码器及依据 USAC WD3 的音频编码器，平均位率的表表示 ；其中图 15 的表显示以帧为基准的 USAC 音频编码器的最小、最大及平均位率的表表示 ；及其中图 16 的表显示基于帧基准的最佳及最恶劣情况的表表示。

[0295] 此外，须注意，依据本发明的实施例提供良好扩充性。通过调整表大小，可依据需求而调整内存需求、运算复杂度、及编码效率间的折衷。

[0296] 9. 位串流语法

[0297] 9.1. 频谱无噪声编码器的有效负载

[0298] 后文中，将叙述有关频谱无噪声编码器的有效负载的若干细节。在若干实施例，有多种不同编码模，诸如所谓的线性预测域、“编码模”及“频域”编码模。在线性预测域编码模，基于音频信号的线性预测分析而执行噪声成形，并且噪声成形信号是在频域被编码。在频域模，基于心理声学分析执行噪声成形，并且音频内容的噪声成形版本在频域中被编码。

[0299] 得自“线性预测域”编码信号及“频域”编码信号二者的频谱系数是经定标量化，然后通过调适性上下文相依性算术编码而以无噪声式编码。量化系数从最低频传输至最高频。各个单独量化系数分裂成最高有效逐 2- 位平面 m，及其余较低有效位平面 r。值 m 是依据该系数的邻近编码。其余较低有效位平面 r 是经熵编码，而未考虑上下文。值 m 及 r 形成算术编码器的符号。

[0300] 算术解码程序的细节描述于此处。

[0301] 9.2. 语法元素

[0302] 后文中，将参考图 6a 至 6h 描述载有算术式编码频谱信息的位串流的位串流语法。

[0303] 图 6a 显示所谓的 USAC 原数据区块(“usac_raw_data_block ()”) 的语法表示。

[0304] USAC 原数据区块包含一个或多个单信道元素(“single_channel_element ()”)和 / 或一个或多个信道对元素(“channel_pair_element ()”)。

[0305] 现在参考图 6b，叙述单信道元素的语法。依据核心模，单信道元素包含线性预测域信道串流(“lpd_channel_stream ()”) 或频域通道串流(“fd_channel_stream ()”)。

[0306] 图 6c 显示信道对元素的语法表示。信道对元素包含核心模信息(“core_mode0”、“core_mode1”)。此外，信道对元素包含配置信息“ics_info ()”。此外，依核心模信息而定，该信道对元素包含与该些信道中的第一个相关联的线性预测域信道串流或频域通道串流，及该信道对元素也包含与该些通道中的第二个相关联的线性预测域信道串流或频域通道串流。

[0307] 配置信息“ics_info ()”的语法表示显示在图 6d，包含多个不同配置信息项，其与本发明并非特别有关。

[0308] 语法表示显示于图 6e 的频域通道串流(“fd_channel_stream ()”)，包含增益信息(“global_gain”)及配置信息(“ics_info ()”)。此外，频域信道串流包含定标因子数据(“scale_factor_data()”)，其描述用于不同定标因子带的频谱值定标的定标因子，并且其被(例如)定标器 150 及复位标器 240 应用。频域信道串流也包含表示算术式编码频谱值的算术式编码频谱数据(“ac_spectral_data ()”)。

[0309] 语法表示显示于图 6f 的算术式编码频谱数据(“ac_spectral_data ()”)，包含用于选择性地重置上下文的选择性算术重置标记(“arith_reset_flag”)，如上所述。此外，算术式编码频谱数据包含多个算术 - 数据区块(“arith_data”)，其载有算术式编码频谱值。该算术式编码数据区块的结构取决于频带数目(以变量“num_bands”表示)，并且也取决于算术重置标记的状态，稍后详述。

[0310] 算术式编码数据区块的结构也将参考图 6g 作说明，该图显示该算术式编码数据区块的语法表示。算术式编码数据区块内部的数据表示是取决于要被编码的频谱值数目 lg、算术重置标记状态、并且还取决于上下文，即，事先解码频谱值。

[0311] 用于频谱值的目前集合编码的上下文是依据参考标号 660 所示的上下文判定运算法则而判定。前文已经参考图 5a 讨论的上下文判定运算法则的细节。算术式编码数据区块包含 lg 个码字组集合，各个码字组集合代表一个频谱值。一个码字组集合包含使用 1 至 20 位表示频谱值的最高有效位平面值 m 的算术码字组“acod_m[pki][m]”。此外，若该频谱值需要比最高有效位平面更多的位平面用于正确表示，则该码字组集合包含一个或多个码字组“acod_r[r]”。码字组“acod_r[r]”表示使用 1 至 20 位间的较低有效位平面。

[0312] 但是，若还需要一个或多个较低有效位平面（除了最高有效位平面值之外）用于频谱值的适当表示，则此是使用一个或多个算术逸出码字组（“ARITH_ESCAPE”）进行信号通知。如此，一般可以说，对一频谱值测定需要多少位平面（最高有效位平面及可能地，一个或多个额外较低有效位平面）。若需一个或多个较低有效位平面，则此是由一个或多个算术逸出码字组“acod_m[pki][ARITH_ESCAPE]”进行信号通知，其是依据目前选定的累积频率表编码，其累积频率表指数是以变量 pki 给定。此外，若有一个或多个算术逸出码字组是包含于该位串流，则上下文经调适，可参考参考标号 664、662。接在该一算术逸出码字组后方，算术码字组“acod_m[pki][m]”包含于该比特流，如参考标号 663 所示，其中 pki 标示目前有效概率模型指数（考虑通过包含算术逸出码字组所导致的上下文调适），及其中 m 标示要被编码或要被解码的频谱值的最高有效位平面值。

[0313] 如前文讨论，任何较低有效位平面的存在导致一个或多个码字组“acod_r[r]”的存在，其各自表示最低有效位平面的一位。一个或多个码字组“acod_r[r]”是依据相对应的累积频率表编码，该累积频率表为恒定且为上下文非相干性。

[0314] 此外，须注意，在各个频谱值的编码后，上下文经更新，如参考标号 668 所示，使得该上下文典型地是与两个随后频谱值的编码不同。

[0315] 图 6h 显示定义算术式编码数据区块语法的定义及辅助元素的图例。

[0316] 综上所述，已经叙述位串流格式，其可以由音频编码器 100 提供，并且其可以由音频解码器 200 评估。算术编码频谱值的位串流被编码，使得其匹配前文讨论的解码运算法则。

[0317] 此外，须注意编码是解码的反向运算，使得其通常假设编码器使用前文讨论的表执行表查询，其近似为对于解码器执行表查询的逆。一般地，了解解码运算法则和 / 或期望的位串流语法的本领域技术人员将容易设计算术编码器，该算术编码器提供在位串流语法所定义的及算术解码器所要求的数据。

[0318] 10. 依据图 21 及 22 的其它实施例

[0319] 后文中，将描述依据本发明的若干其它简化实施例。

[0320] 图 21 显示依据本发明的一实施例的一种音频编码器 2100 的方块示意图。该音频编码器 2100 被配置为接收一输入的音频信息 2110，以及基于此信息而提供一已编码的音频信息 2112。该音频编码器 2100 包含一能量压缩时域至频域变换器，其被配置为接收该输入的音频表示 2110 的时域表示 2122，以及基于此表示而提供一频域音频表示 2124，使得该频域音频表示包含一频谱值集合（例如频谱值 a）。该音频编码器 2100 也包含一算术编码器 2130，其被配置为使用一可变长度码字组而编码频谱值 2124 或其预处理版本。该算术编码器 2130 被配置为将一频谱值或一频谱值的最高有效位平面值映射至一码值（例如表示可变长度码字组的一码值）。

[0321] 该算术编码器包含映射规则选择 2132 及一上下文值判定 2136。该算术编码器被配置为依据描述上下文状态的一数值型目前上下文值而选择映射规则，该映射规则描述一频谱值 2124 或一频谱值 2124 的最高有效位平面的映射至一码值（其可表示一可变长度码字组）。该算术解码器被配置为依据多个先前已编码的频谱值而用来判定用在该映射规则选择 2132 的一数值型目前上下文值 2134。该算术编码器或更明确言之，该对映规则的选择 2132 被配置为使用一迭代区间大小缩减而评估至少一个表，来判定该数值型目前上下文值

2134 是否与由该表的登录项目所描述的表上下文值相同，或是否落入在由该表的登录项目所描述的一区间内部，由此而导出描述一经选定的映射规则的映射规则指数值 2133。如此，依据该数值型目前上下文值 2134 能够以高运算效率而选择该映射关系 2131。

[0322] 图 22 显示依据本发明的另一实施例的一种音频信号解码器 2200 的方块示意图。该音频信号解码器 2200 被配置为接收已编码的音频信息 2210，及基于此而提供已解码的音频信息 2212。该音频信号解码器 2200 包含一算术解码器 2220，其被配置为接收该频谱值的已经算术编码的表示 2222，以及基于此而提供多个已解码的频谱值 2224（例如已解码的频谱值 a）。该音频信号解码器 2200 也包含一频域至时域变换器 2230，其被配置为接收该些已解码的频谱值 2224，及使用该些已解码的频谱值而提供一时域音频表示，从而获得该已解码的音频信息 2212。

[0323] 该算术解码器 2220 包含映射关系 2225，其是用来将一码值（例如提取自表示该已编码的音频信息一位串流的一码值）映射至一符号码（该符号码例如可描述一已解码的频谱值或该已解码的频谱值的最高有效位平面）。该算术解码器进一步包含映射规则选择 2226，其提供映射规则选择信息 2227 至该映射关系 2225。该算术解码器 2220 也包含一上下文值判定 2228，其提供一数值型目前上下文值 2229 至该映射规则选择 2226。

[0324] 算术解码器 2220 被配置为依据上下文状态而选定映射规则，该映射规则描述一码值（例如，提取自表示该已编码的音频信息一位串流的一码值）映射至一符号码（例如，表示该已解码的频谱值的一数值，或表示该已解码的频谱值的最高有效位平面的一数值）。该算术解码器被配置为依据多个事先解码频谱值而判定描述该目前上下文状态的一数值型目前上下文值。此外，该算术解码器（或更明确言之，该映射规则选择 2226）被配置为使用迭代区间大小缩减而评估至少一个表，判定该数值型目前上下文值 2229 是否与由该表的一登录项目所描述的表上下文值相同，或是否落在由该表的该些登录项目所描述的区间内部，由此导出描述一选定的映射规则的映射规则指数值 2227。如此，应用于该映射关系 2225 的映射规则能够以运算有效方式选择。

[0325] 11. 实施替代例

[0326] 虽然在装置的上下文已经描述若干方面，但显然此些方面也表示相对应方法的说明，此处区块或装置与方法步骤或方法步骤的特征相对应。类似地，在方法步骤的上下文所述方面也表示相对应区块或相对应装置的项目或特征的描述。部分或全部方法步骤可由（或使用）硬件装置执行，例如微处理器、可程序计算机或电子电路。在若干实施例中，最重要方法步骤中的某一个或多个可由此种装置执行。

[0327] 本发明编码的音频信号可储存于数字储存介质，或可在传输媒介上传输，诸如无线媒介媒体或有线传输媒介（诸如，互联网）。

[0328] 依据某些实施要求而定，本发明实施例可以硬件或软件实施。实施可使用具有可电子式读取的控制信号储存其上的数字储存介质，例如软盘、DVD、蓝光盘、CD、ROM、PROM、EPROM、EEPROM 或闪存执行，该些控制信号与可程序计算机协力合作，使得可执行各个方法。因此，数字储存介质可以是计算机可读的。

[0329] 依据本发明的若干实施例包含一数据载体，其具有可电子式读取的控制信号，该些控制信号与可程序计算机协力合作，使得可执行此处所述方法中的一个。

[0330] 一般而言，本发明的实施例可实施为带有程序代码的计算机程序产品，当该计算

机程序产品运行在计算机上时，该程序代码可操作地执行该方法中的一个。程序代码例如可储存在机器可读取载体上。

[0331] 其它实施例包含用以执行储存在机器可读取载体上的此处所述方法中的一个的计算机程序。

[0332] 换言之，因此，本发明方法的实施例为具有程序代码用以执行储存在机器可读取载体上的此处所述方法中的一个的计算机程序。

[0333] 因此，本发明方法的又一实施例为数据载体(或数字储存介质、或计算机可读介质)包含用以执行此处所述方法中的一个的计算机程序记录于其上。

[0334] 因此，本发明方法的又一实施例为一数据串流或一序列信号，表示用以执行此处所述方法中的一个的计算机程序。该数据串流或信号序列例如可被配置为经由数据通讯连接(例如，经由互联网)而传输。

[0335] 又一实施例包含被配置为或被调适为执行此处所述方法中的一个的处理装置，例如计算机或可编程逻辑装置。

[0336] 又一实施例包含其上已经安装计算机程序用以执行此处所述方法中的一个的计算机。

[0337] 在若干实施例，可编程逻辑装置(例如，现场可编程门阵列)可用来执行此处所述方法的部分或全部功能。在若干实施例，现场可编程门阵列可与微处理器协力合作来执行此处所述方法中的一个。大致上，该等方法较佳是由任何硬件装置执行。

[0338] 前述实施例仅供举例说明本发明的原理。须了解，此处所述配置及细节的修正与变更对于本领域的技术人员是显而易见的。因此，本发明的范围仅受所附权利要求书的范围限制，而非受此处实施例的描述及解说所呈现的特定细节所限。

[0339] 虽然前文已经特别显示及参考前述特定实施例作说明，但本领域技术人员须了解在未背离其精神及范围的情况下，可以在形式与细节上作出多项其它改变。须了解，在未背离本文所公开的及随后的权利要求包含的广义概念的情况下，适应于不同实施例而做出多项变化。

[0340] 12. 结论

[0341] 总结而言，发现依据本发明的实施例形成一种改良式无噪声编码方案。依据该新颖提案的实施例允许将内存需求自 16894.5 字组减少至 900 字组(ROM)及自 666 字组减少至 72 字组(每个核心编码器信道的静态 RAM)。如此，允许在一个实施例中的完整系统的数据 ROM 需求减少约 43%。同时，不仅完全维持编码性能，同时甚至平均增高编码性能。WD3 的(依据 USAC 草拟标准 WD3 所提供的位串流的)无损耗转码被证实为可能。如此，通过将此处所述无噪声编码采用至该 USAC 草拟标准的未来工作草案，获得依据本发明的实施例。

[0342] 要言之，在一实施例，所提出的新颖无噪声编码可导致 MPEG USAC 草拟标准就下列方面的修正：就如图 6g 所示位串流元素“arith_data ()”的语法；就前述频谱无噪声编码器的有效负载且如图 5h 所示；就前述频谱无噪声编码；就如图 4 所示的状态计算的上下文；就如图 5i 所示的定义；就前文参考图 5a、5b、5c、5e、5g、5h 所述的解码程序；及就如图 17、18、20 所示的表；及就如图 5d 所示的函数“get_pk”。但另外，依据图 20 的表“ari_s_hash”可用来替代图 17 的表“ari_s_hash”，及图 5f 的函数“get_pk”可用来替代依据图 5d 的函数“get_pk”。

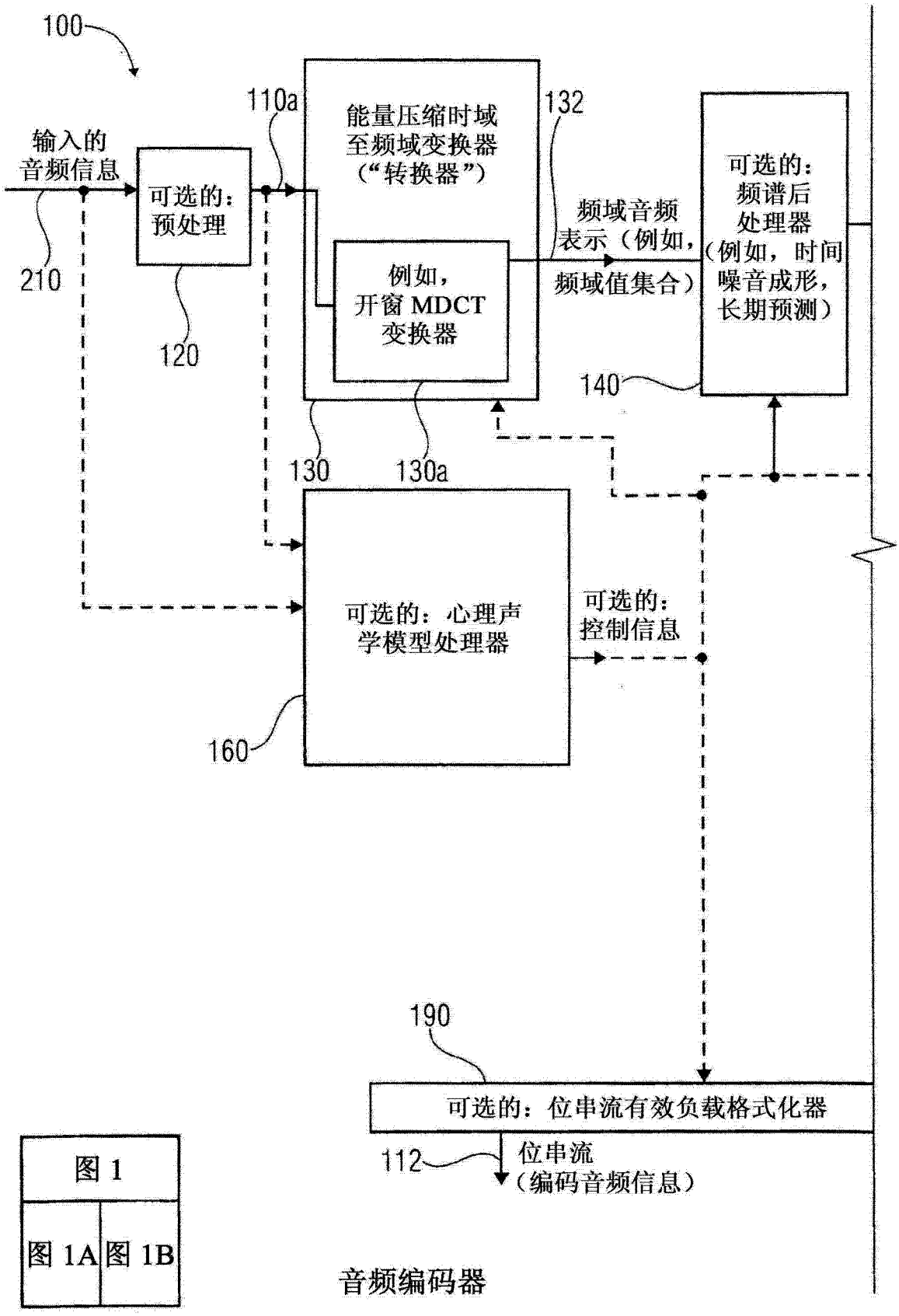


图 1	
图 1A	图 1B

图 1A

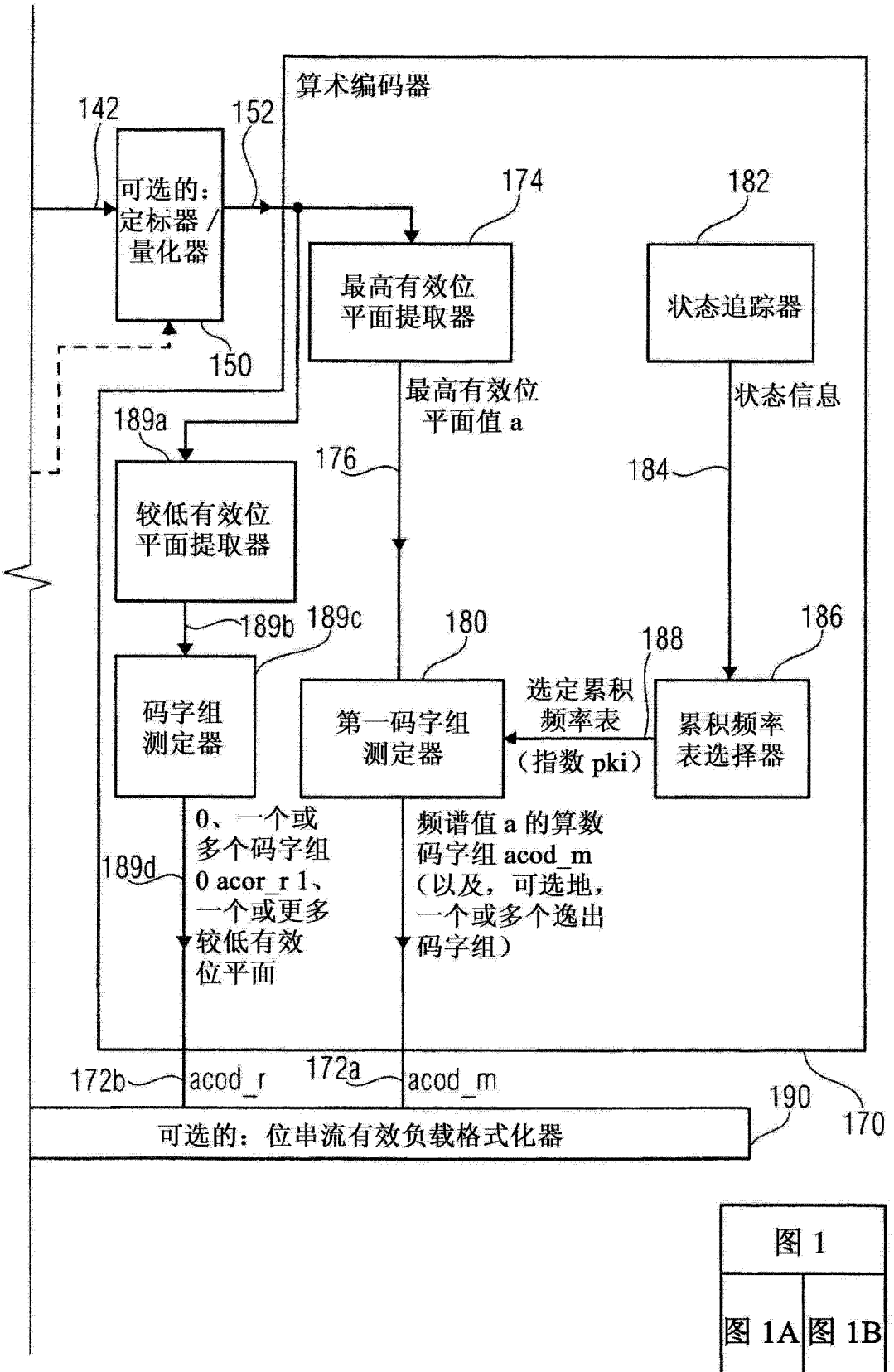


图 1

图 1A 图 1B

音频编码器

图 1B

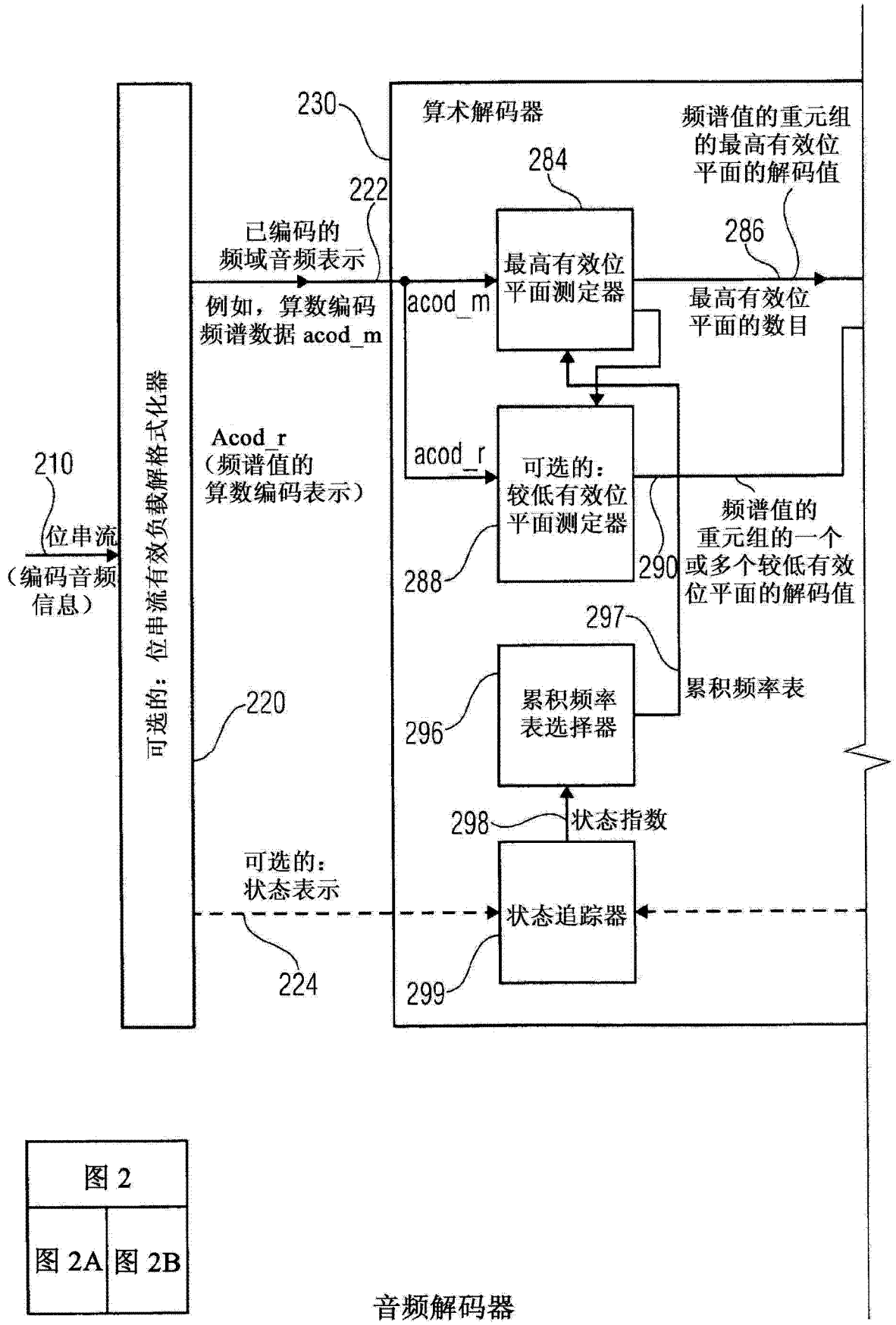


图 2

图 2A 图 2B

图 2A

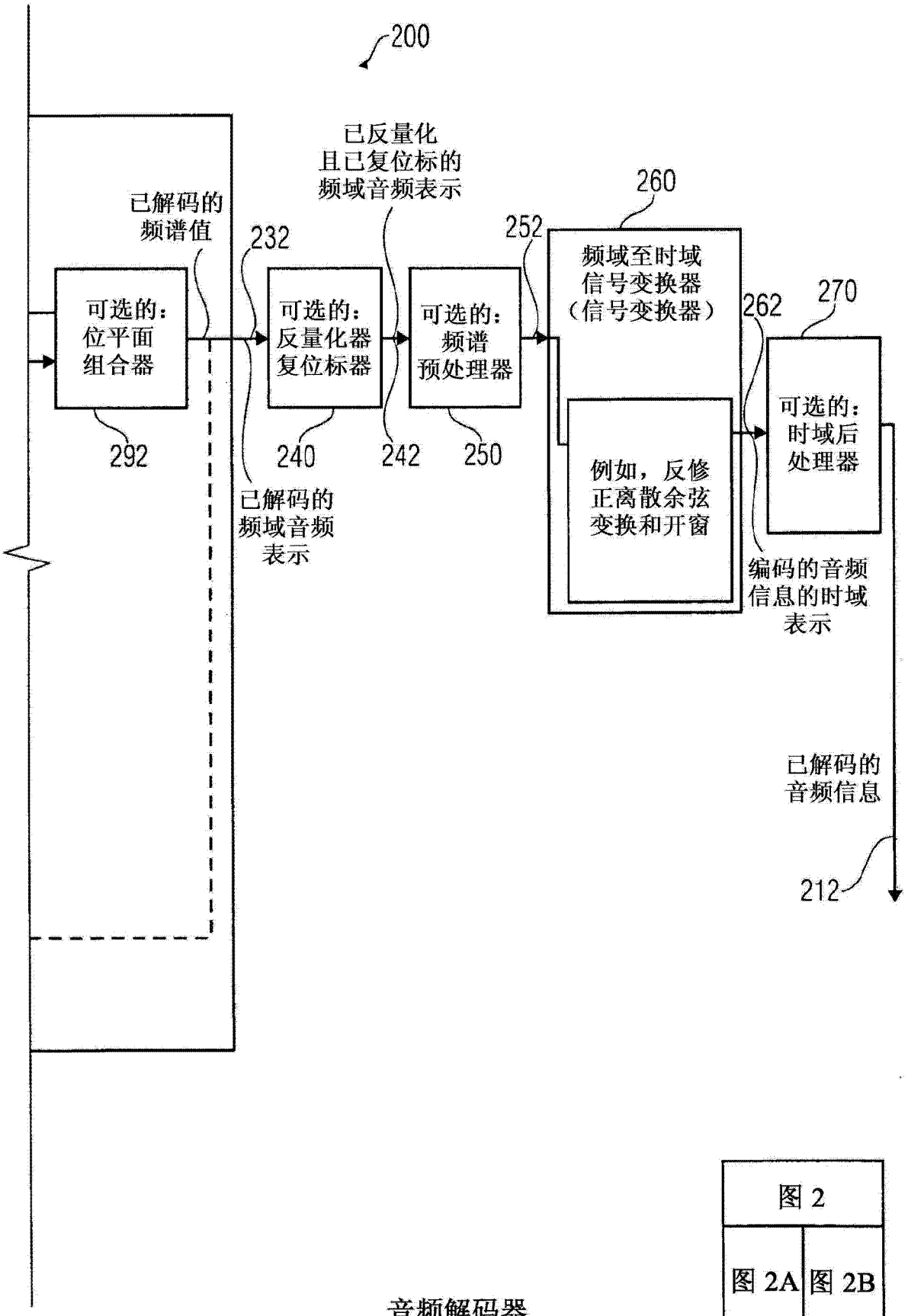


图 2

图 2A 图 2B

图 2B

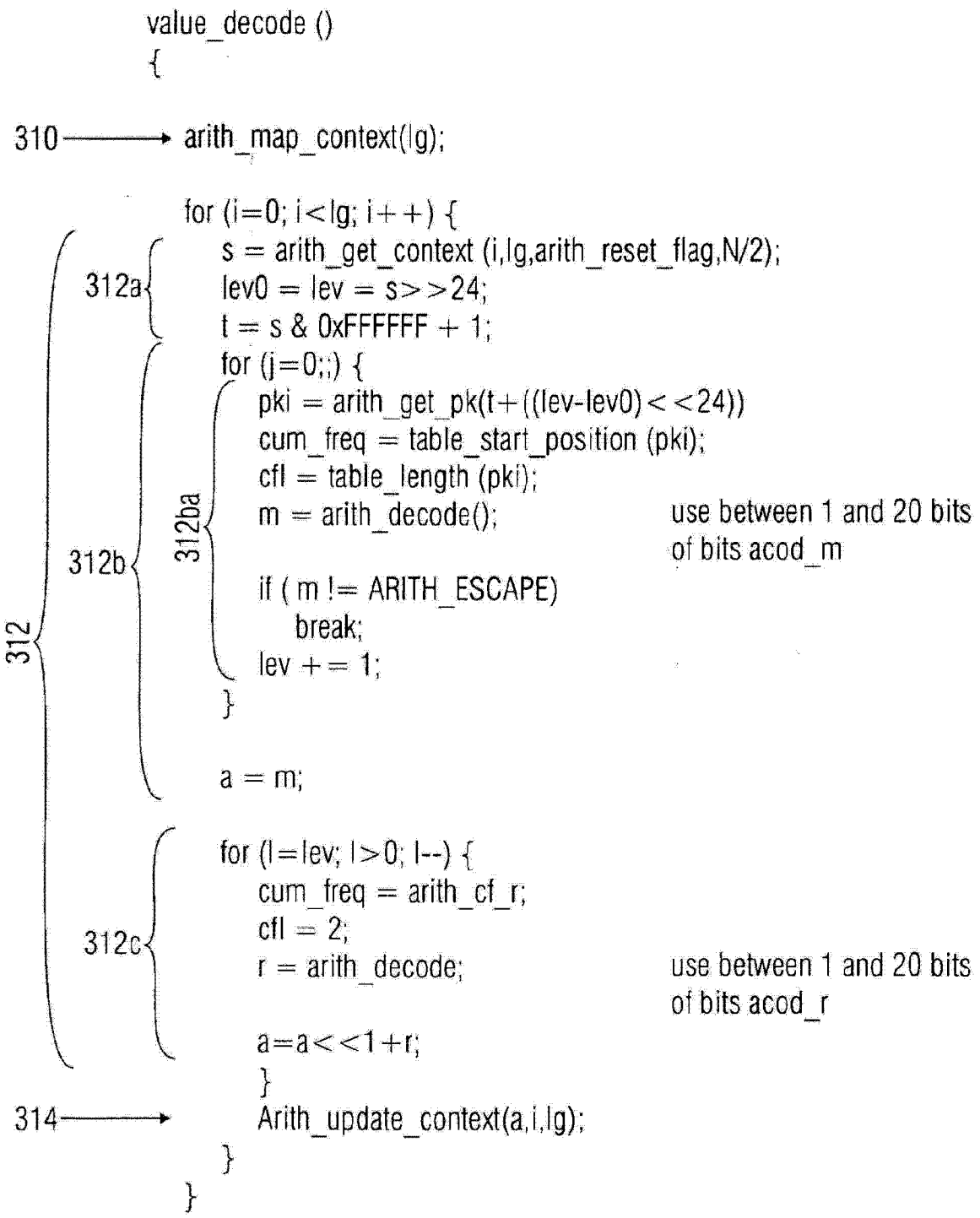

value_decode ()
{
310
arith_map_context(lg);
for (i=0; i<lg; i++) {
312a
s = arith_get_context (i,lg,arith_reset_flag,N/2);
lev0 = lev = s>>24;
t = s & 0xFFFFFF + 1;
312b
for (j=0;;) {
312ba
pki = arith_get_pk(t+((lev-lev0)<<24))
cum_freq = table_start_position (pki);
cfl = table_length (pki);
m = arith_decode();
use between 1 and 20 bits
of bits acod_m
if (m != ARITH_ESCAPE)
break;
lev += 1;
}
a = m;
312
312c
for (l=lev; l>0; l--) {
cum_freq = arith_cf_r;
cfl = 2;
r = arith_decode;
use between 1 and 20 bits
of bits acod_r
a=a<<1+r;
}
314
Arith_update_context(a,i,lg);
}
}

图 3

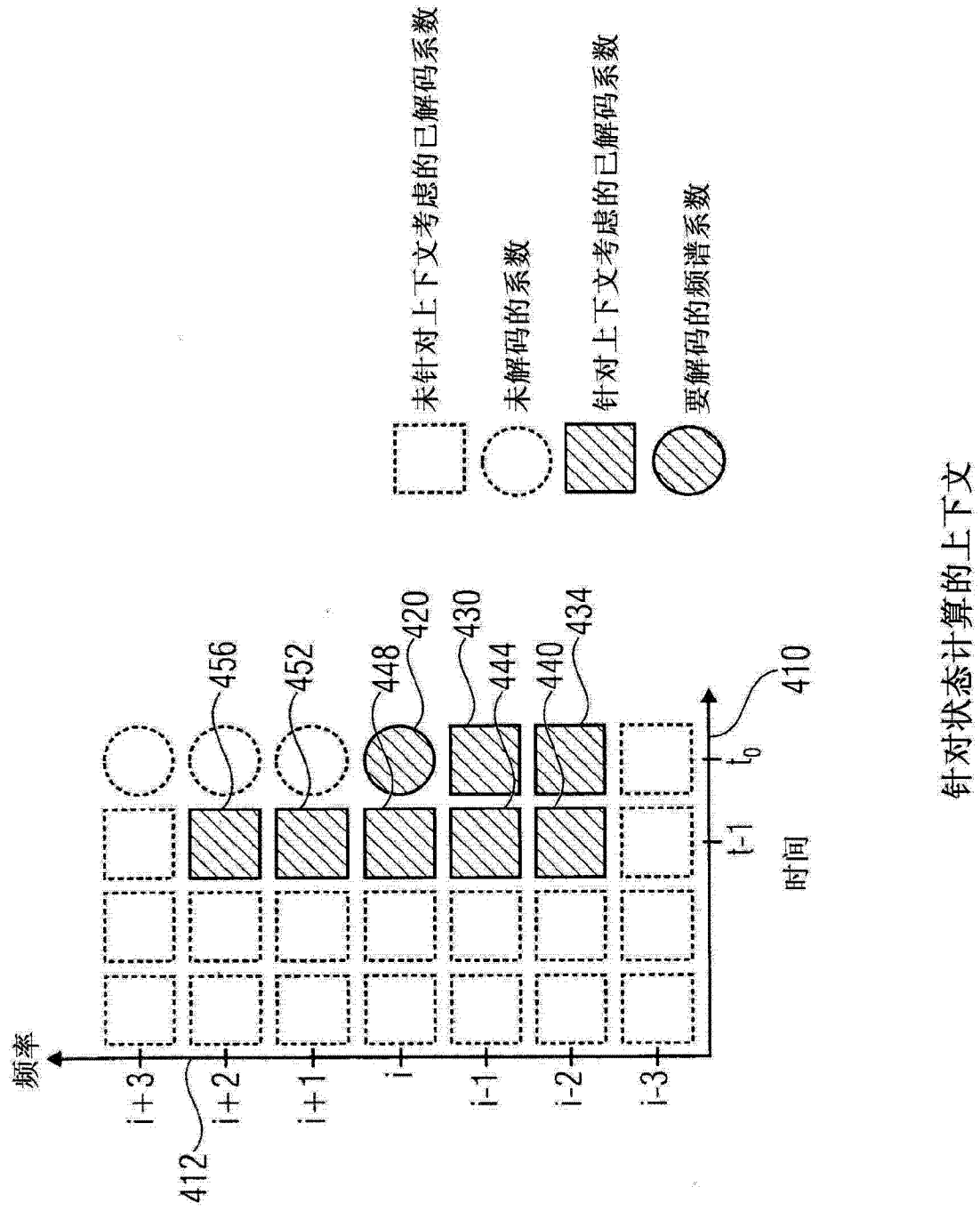

未针对上下文考虑的已解码系数
未解码的系数
针对上下文考虑的已解码系数
要解码的频谱系数
频率
时间
i+3
i+2
i+1
i
i-1
i-2
i-3
412
456
452
448
420
430
444
440
434
410
t-1
t_0
针对状态计算的上下文

图 4

```
/*Input variables*/
lg /*number of sepctral coefficients to decode in the frame*/
previous_lg /Previous number of spectral lines of the previous frame*/

arith_map_context()
{
      v=w=0

      ratio= ((float)previous_lg)/((float)lg);
      for(j=0; j<lg; j++){
            k = (int) ((float)) ((j)*ratio);
            q[0][v++].c = qs[w+k];
      }

      previous_lg=lg;
}
```

图 5A

```
/*Input variables*/
i  /*Index of the spectral value to decode in the vector*/
lg /*Number of expected quantized coefficients*/
N  /*Number of lines of the transformation*/
ari_reset_flag  /*flag indicating whether the context should be reset*/
/*Output value*/
t  /*Concatenated state index s and predicted bit-plane level lev0*/

arith_get_context()
{
    int a0,c0,c1,c2,c3,c4,c5,c6,lev0,region;

    if(arith_reset_flag && i==0)
        return(0);
    if((!arith_reset_flag) && (i!=0)){
        int k;
        int lim_min,lim_max;
        int flag=1;
        lim_max = i+6;
        if((i+lim_max)>lg-1)
            lim_max=lg-1-i;
        lim_min = -5;
        if((i+lim_min)<0)
            lim_min=-i;
        for(k=lim_min;k<0;k++)
            if(q[0][k].c!=0 && q[1][k].c!=0)
                flag=0; break;
        for(;k<=lim_max;k++)
            if(q[0][k].c!=0)
                flag=0; break;
        if(flag)
            return(1);
    }
    if(i>0){
        a0=q[1][i-1];
        c0=ABS(a0);
        lev0=0;
        while((a0<-4)||(a0>=4)){
            a0>>=1;
            lev0++;
            c0=4+lev0;
        }
        if(c0>7)
            c0=7;
        if(lev0>3)
            lev0=3;

        if(arith_reset_flag && i==1)
            return((2+c0)|(lev0<<24));
        c4=q[0][i-1].c;
    }
```

510 512 512a 512b 512c 512d 514 514a 514b 514c 514d 514e

<接续图 5C>

图 5B

<接续图 5B>

```
516 {  if(i>1){
           c1=q[1][i-2].c;
           lev0=MAX(q[1][i-2].l,lev0);
           c6=q[0][i-2].c;
       }
       if(i>2){
518 ---->  lev0=MAX(q[1][i-3].l,lev0);
       }
520 {      if(i<N/4)
               region=0;
           else if(i<N/2)
               region=1;
           else
               region=2;
       }

522 {  if(i>3)
           lev0=MAX(q[1][i-4].l,lev0);
524 {  if(lev0>3)
           lev0=3;
526 {  if(arith_reset_flag)
           return((10+4*(8*c0+c1)+region)|(lev0<<24));

528 ---->  c2=q[0][i].c;
       }
530 {      if(i<lg-1)
           c3=q[0][i+1].c;
       else
           c3=0;
       }
532 {  if(i<lg-2)
           c5=q[0][i+2].c;
       else
           c5=0;
       }

534 {  if(lev0==0)
           if((c2==3 || c3 ==3) &&i==0)
               lev0=1;
       }

536 {  if(i==0)
536a ---->     return((249+4*(4*c2+c3)+c5)|(lev0<<24));
       else if(i==1)
536b ---->     return((313+4*(4*(4*(8*c0+c2)+c3)+c4)+c5)|(lev0<<24));
       else
536c ---->     return((4212+4*(4*(4*(4*(4*(4*(8*c0+c2)+c3)+c4)+c1)+c5)+c6)+region)
                       |(lev0<<24));
   }
```

图 5C

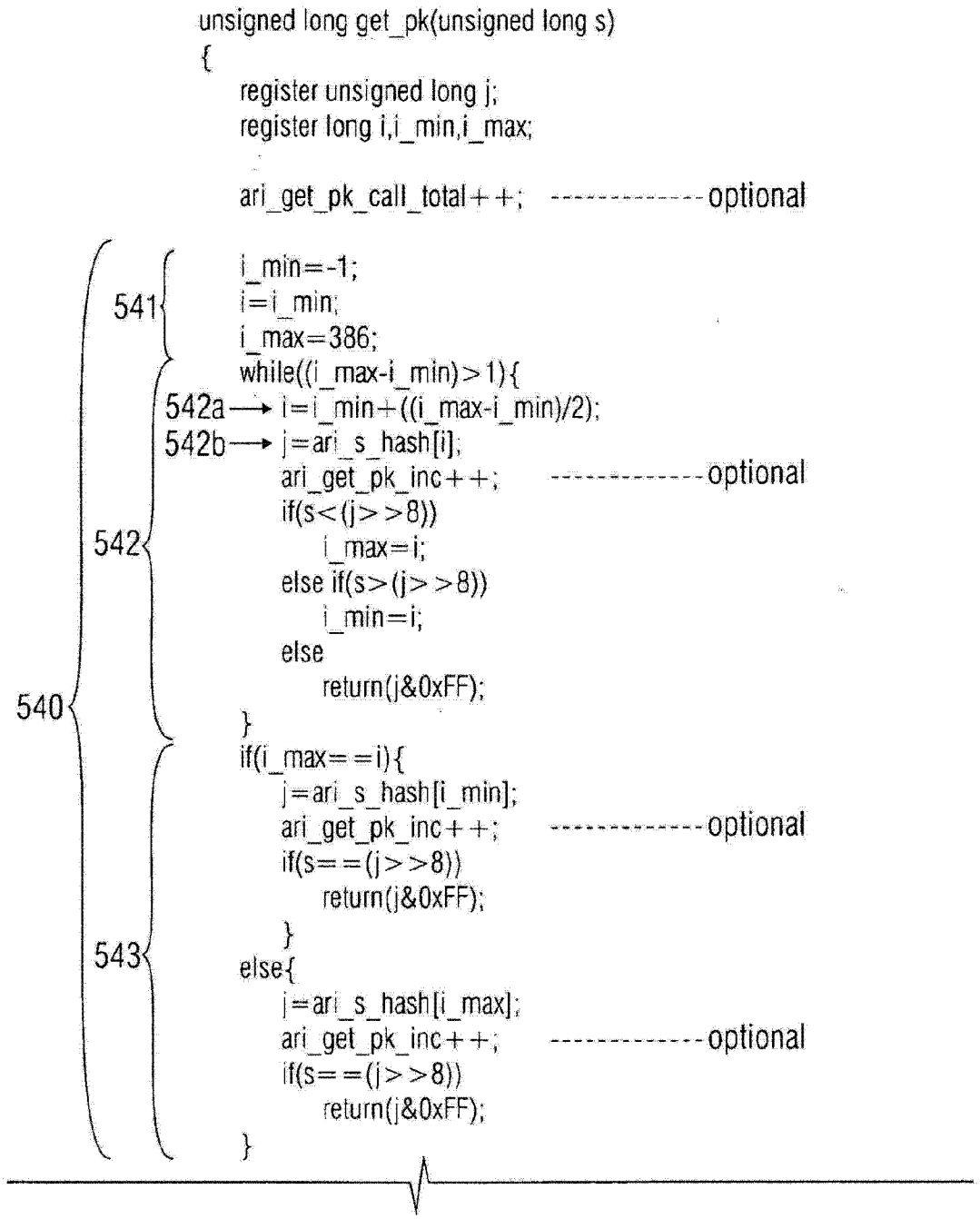


图 5D1	图 5D
图 5D2	

图 5D1

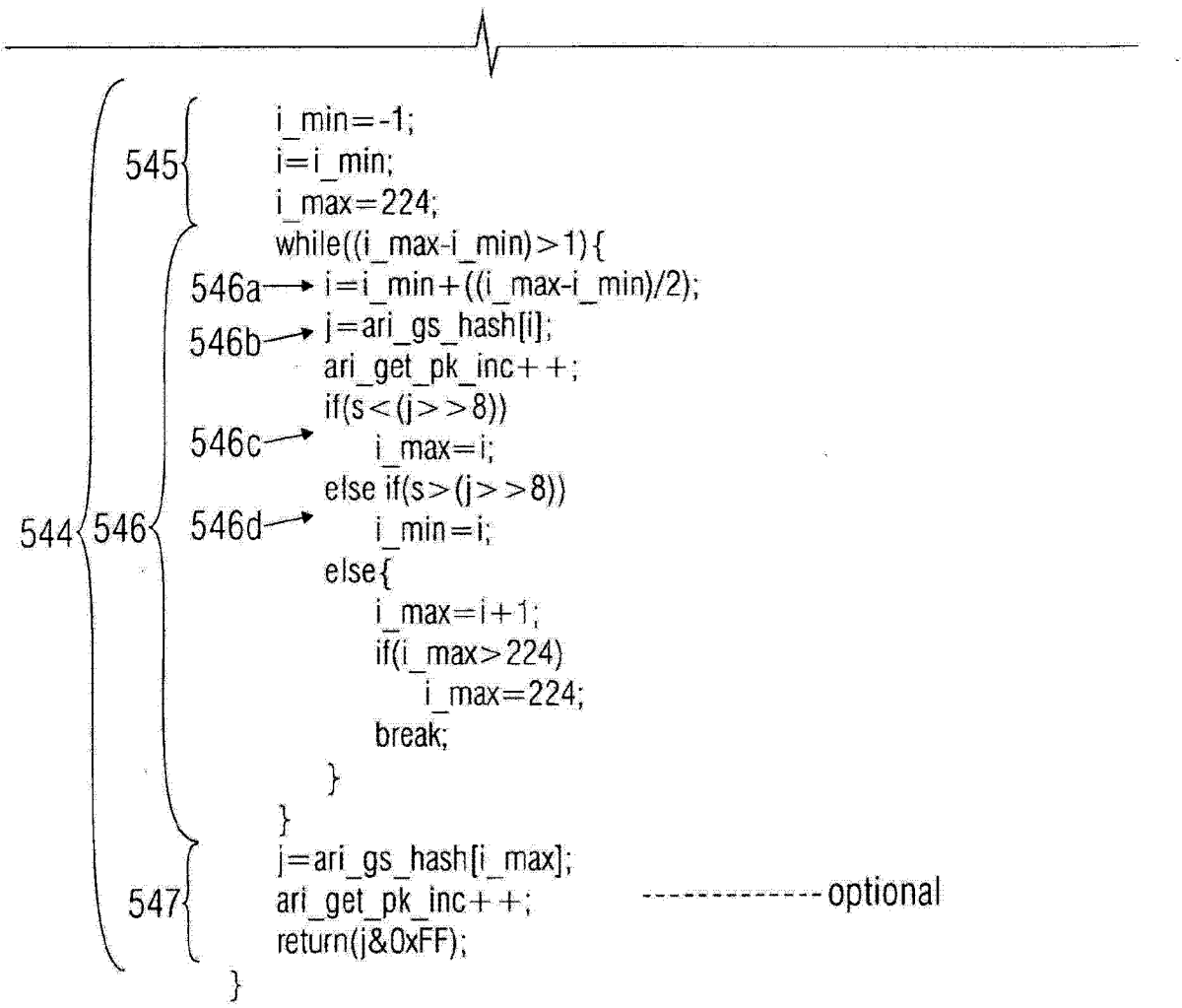


图 5D1	图 5D
图 5D2	

图 5D2

```
/*Input variable*/
s  /*State of the context*/
/*Output value*/
pki /*Index of the probability model */

arith_get_pk(s)
{
    register unsigned long i,j;

    for (i=0;i<387;i++)
    {
        j=ari_s_hash[i];
        if ( (j>>8)==s ) return j&255;
    }
    for (i=0;i<225;i++)
    {
        j=ari_gs_hash[i];
        if ( s<(j>>8) ) return j&255;
    }
    return j&255;
}
```

图 5E

```
unsigned long get_pk(unsigned long s)
{
    register unsigned longlong j;
    register unsigned long  i;

    for (i=0;i<387;i++)
    {
        j=ari_s_hash[i];
        if ( (j>>8)==s )
            return j&0xFF;
    }
    for(i=0;i<225;i++){
        j=ari_gs_hash[i];
        if ( s<(j>>8) ) return j&0xFF;
    }
    return(j&0xFF);
}
```

图 5F

```
/*helper funtions*/
bool arith_first_symbol(void);
    /* Return TRUE if it is the first symbol of the sequence, FALSE otherwise*/
Ushort arith_get_next_bit(void);
    /* Get the next bit of the bitstream*/

/* global variables */
low
high
value

/* Input variables */
cum_freq[];  /* cumulative frequencies table*/
cfl;             /* length of cum_freq[] */

arith_decode()
{
    if(arith_first_symbol())
    {
        value = 0;
        for (i=1; i<=20; i++)
        {
        value = (val<<1) | arith_get_next_bit();
        }
        low=0;
        high=1048575;
    }

    range = high-low+1;
    cum =((((int64) (value-low+1))<<16)-((int64) 1))/((int64) range);
    p = cum_freq-1;

    do
    {
        q=p+(cfl>>1);
        if ( *q > cum ) {p=q; cfl++; }
        cfl>>=1;
    }
        while ( cfl>1 );
```

570a 570b 570c

图 5G1	图 5G
图 5G2	

图 5G1

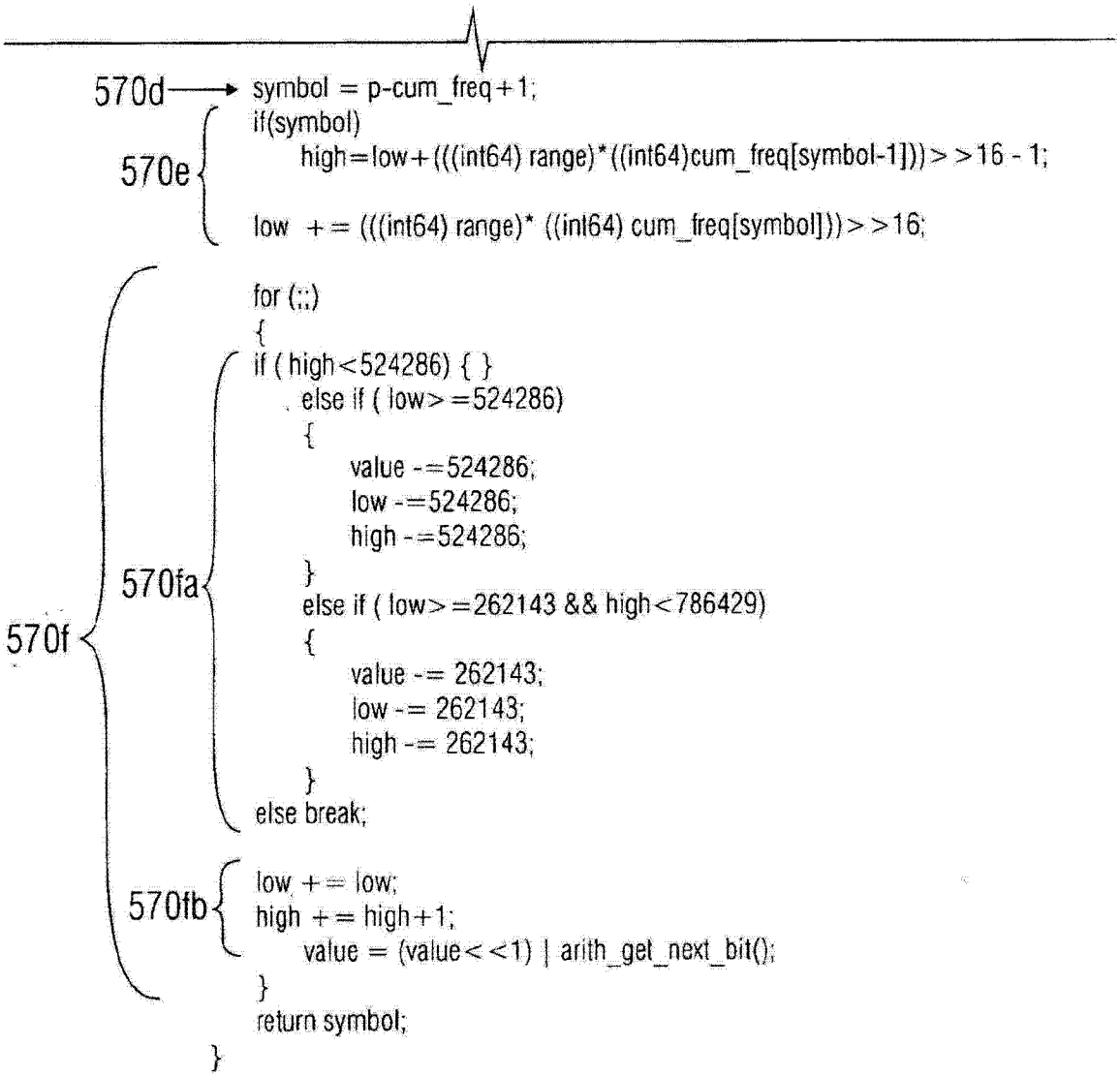


图 5G1	图 5G
图 5G2	

图 5G2

```
/*input variables*/
a /*Decoded quantized spectral coefficient */
i /*Index of the quantized spectral coefficient to decode*/
lg /*number of coefficients in the frame*/

arith_update_context()
{   int a0;

580 ──► q[1][i]=a0=a;
        q[1][i].l=0;
        while(ABS(a0)>4){
582         a0=a0>>1;
            q[1][i].l++;
        }
        if(a==0)
            q[1][i].c=0;
        else if(ABS(a)<=1)
            q[1][i].c=1;
584     else if(ABS(a)<=3)
            q[1][i].c=2;
        else
            q[1][i].c=3;

        if(i==lg && core_mode==1){
            ratio= ((float) lg)/((float)1024);
            for(j=0; j<1024; j++){
586             k = (int) ((float) j*ratio);
                qs[j] = q[1][k].c ;
            }
            previous_lg = 1024;
        }
        if(i==lg/4 && core_mode==0){
            for(j=0; j<MIN(lg,1024; j++){
588             qs[j] = q[1][j].c;
            }
            previous_lg = MIN(1024,lg);
        }
}
```

图 5H

定义

a	要解码的量化系数
m	要解码的量化频谱系数的最高有效逐 2- 位平面
r	要解码的量化频谱系数的最高有效逐 2- 位平面
lev	剩余位平面的位准。其对应于较低有效比最高有效逐 2- 位平面的数量
lev0	预测位平面位准
arith_s_hash[]	上下文状态映射至累积频率表系数 pki 的哈希表
arith_gs_hash[]	上下文状态组映射至累积频率表系数 pki 的哈希表
arith_cf_m[pki][9]	针对最高有效逐 2- 位平面 m 和 ARITH_ESCAPE 符号的累积频率的模型
arith_cf_r []	针对较低有效位平面符号 r 的累积频率
previous_lg	由算数解码器事先解码的所传输的频谱系数的数目
N	开窗长度。针对 AAC，从 window_sequence（见 6.8.3.1 节）及针对 TCX N=2.1g 导出。
q[2][]	针对要解码的频谱系数的当前上下文
qs[]	针对下一帧所存储的过去上下文
arith_reset_flag	指示是否必须重置频谱无噪声上下文的标记

图 5I

```
usac_raw_data_block ()
{
    single_channel_element (); and/or
    channel_pair_element ();
}
```

图 6A

Syntax of single_channel_element()

语法	位数目	Mnemonic
single_channel_element() {		
core_mode	**1**	**uimsbf**

if (core_mode == 1) {
lpd_channel_stream();
}
else {
fd_channel_stream();
}		

}		

图 6B

Syntax of channel_pair_element()

语法	位数目	Mnemonic
channel_pair_element()		
{		
core_mode0	**1**	**uimsbf**
core_mode1	**1**	**uimsbf**
ics_info();	optional: common ics_info for two channels	
if (core_mode0 == 1) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
if (core_mode1 == 1) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
}		

图 6C

Syntax of ics_info()

语法	位数目	Mnemonic
ics_info()		
{		
window_length;	1	uimsbf
if(window_length != 0) {		
transform_length;	1	uimsbf
}		
else {		
transform_length = 0;		
}		
window_shape;	1	uimsbf
if (window_length != 0 && transform_length != 0){		
max_sfb;	4	uimsbf
scale_factor_grouping;	7	uimsbf
}		
else {		
max_sfb;	6	uimsbf
}		
}		

可选的

图 6D

Syntax of fd_channel_stream()

语法	位数目	Mnemonic
fd_channel_stream() {		
global_gain;	**8**	**uimsbf**
ics_info(); (unless included in channel pair element)		
scale_factor_data ();		
ac_spectral_data ();		
}		

图 6E

Syntax of ac_spectral_data()

语法	位数目	Mnemonic
ac_spectral_data() {		
arith_reset_flag	**1**	**uimsbf**
for (win=0; win<num_windows; win++){		
arith_data(num_bands, arith_reset_flag)		
}		
}		

图 6F

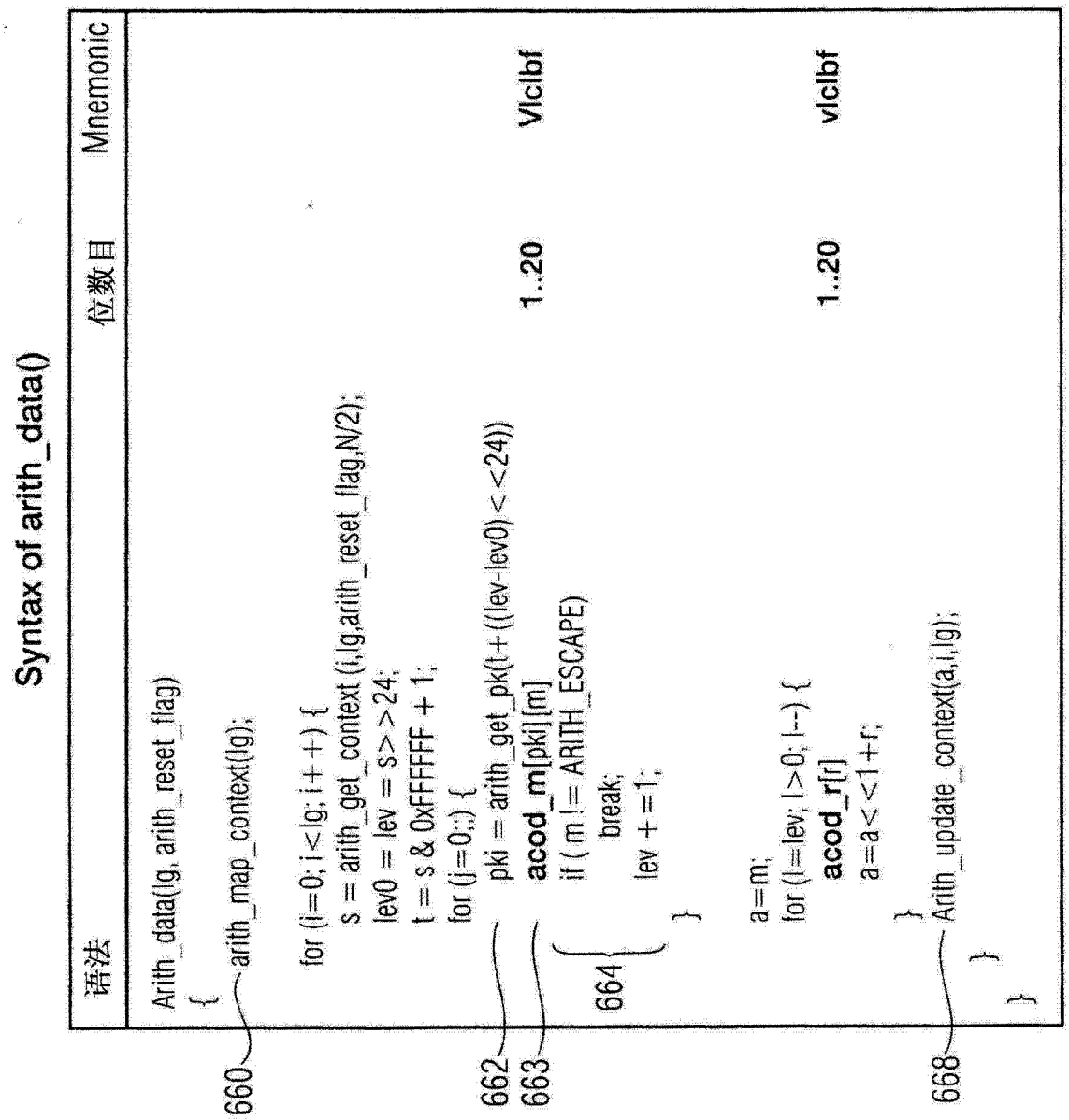

Syntax of arith_data()
语法
位数目
Mnemonic
Arith_data(lg, arith_reset_flag)
{
arith_map_context(lg);
for (i=0; i<lg; i++) {
s = arith_get_context (i,lg,arith_reset_flag,N/2);
lev0 = lev = s>>24;
t = s & 0xFFFFFF + 1;
for (j=0;;) {
pki = arith_get_pk(t+((lev-lev0)<<24))
1..20
VlclbfacodM[pki][m]
if (m != ARITH_ESCAPE)
break;
lev += 1;
}
a=m;
for (l=lev; l>0; l--) {
acodR[r]
1..20
vlclbf
a=a<<1+r;
}
Arith_update_context(a,i,lg);
}
}
660
662
663
664
668

图 6G

定义

arith_data()	要解码频谱无噪声解码器数据的数据元素
arith_reset_flag	指示是否必须重置频谱无噪声上下文的标记
acod _cf_m[pki][a]	针对量化频谱系数的最高有效逐 2- 位平面的算数解码所需的算数码字组
arith_cf_r[]	针对量化频谱系数的其余位平面的算数解码所需的算数码字组
辅助元素	
a	要解码的量化系数
m	要解码的量化频谱系数的最高有效逐 2- 位平面
r	要解码的量化频谱系数的最高有效逐 2- 位平面
N	开窗长度。针对 AAC，从 window_sequence（见 6.8.3.1 节）及针对 TCX N=2.1g 导出。
lg	要解码的量化系数的数目
i	在帧内要解码的量化系数的指数
pki	由算数解码器用于解码 a 所使用的累积频率表的指数
arith_get_pk ()	返回解码码字组 acod_ng[pki][a] 所需的累积频率表的指数 pki 的函数
t	上下文状态
arith_get_context ()	返回上下文的状态的函数
lev0	预测位平面位准
s	与预测位平面位准值 lev0 结合的上下文状态
lev	超出最高有效逐 2- 位平面的要解码的位平面的位准值
ARITH_ESCAPE	指示超出预测位平面位准值 lev0 的要解码的附加位平面的逸出符号

图 6H

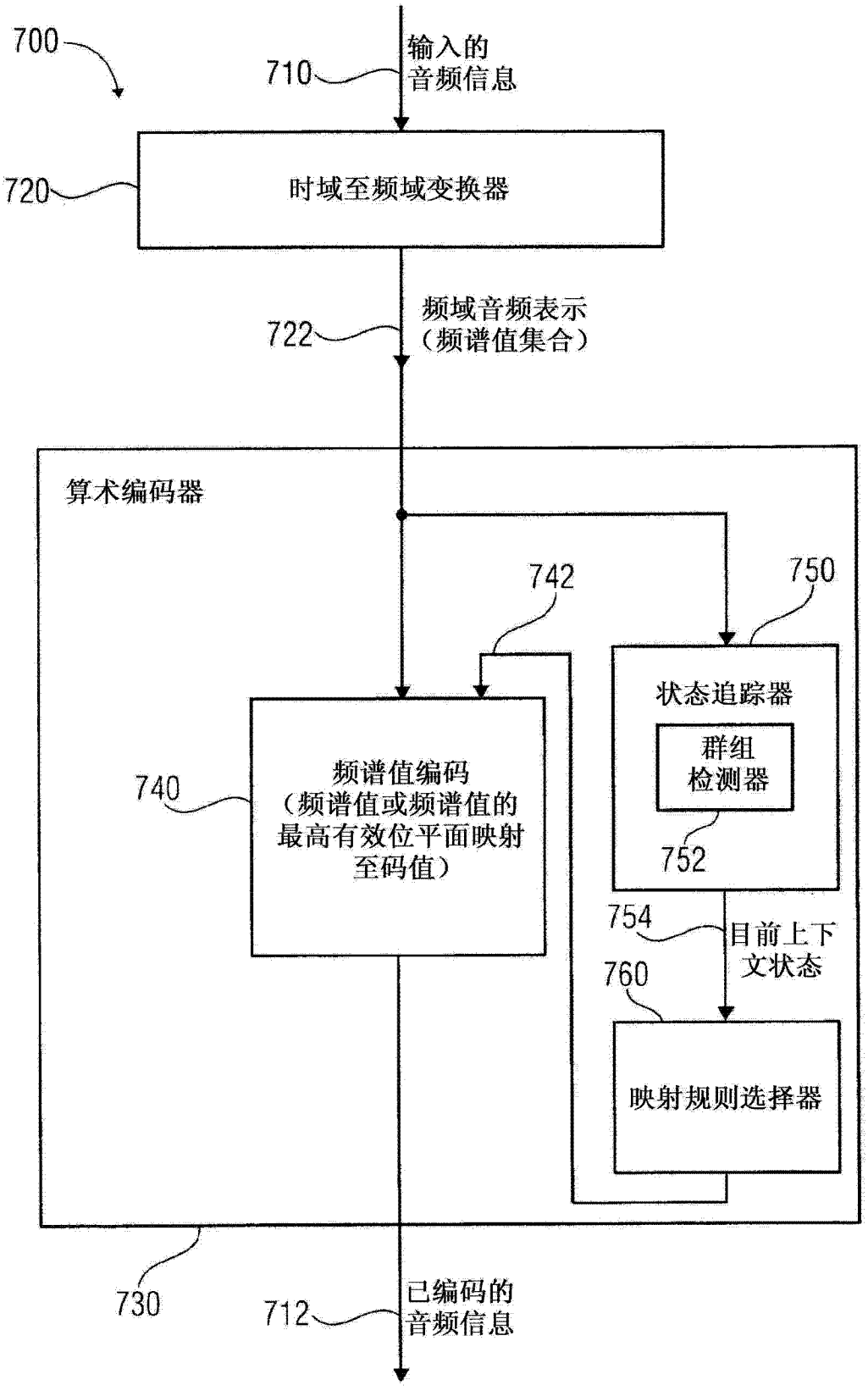

700
710
输入的
音频信息
720
时域至频域变换器
722
频域音频表示
（频谱值集合）
算术编码器
742
750
状态追踪器
群组
检测器
752
740
频谱值编码
（频谱值或频谱值的
最高有效位平面映射
至码值）
754
目前上下
文状态
760
映射规则选择器
730
712
已编码的
音频信息

图 7

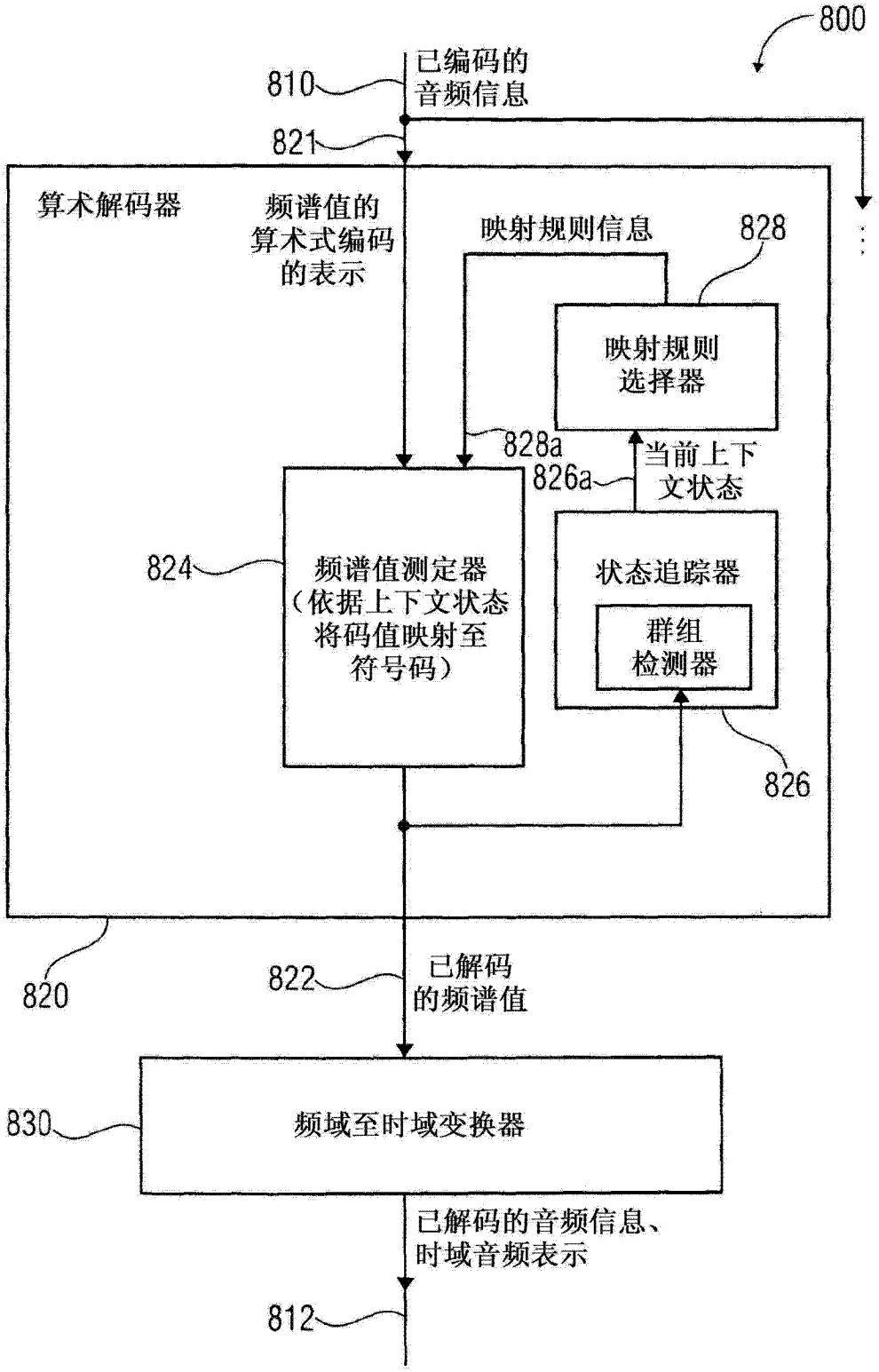

800
已编码的
音频信息
810
821
算术解码器
频谱值的
算术式编码
的表示
映射规则信息
828
映射规则
选择器
828a
826a
当前上下
文状态
824
频谱值测定器
（依据上下文状态
将码值映射至
符号码）
状态追踪器
群组
检测器
826
820
822
已解码
的频谱值
830
频域至时域变换器
已解码的音频信息、
时域音频表示
812

图 8

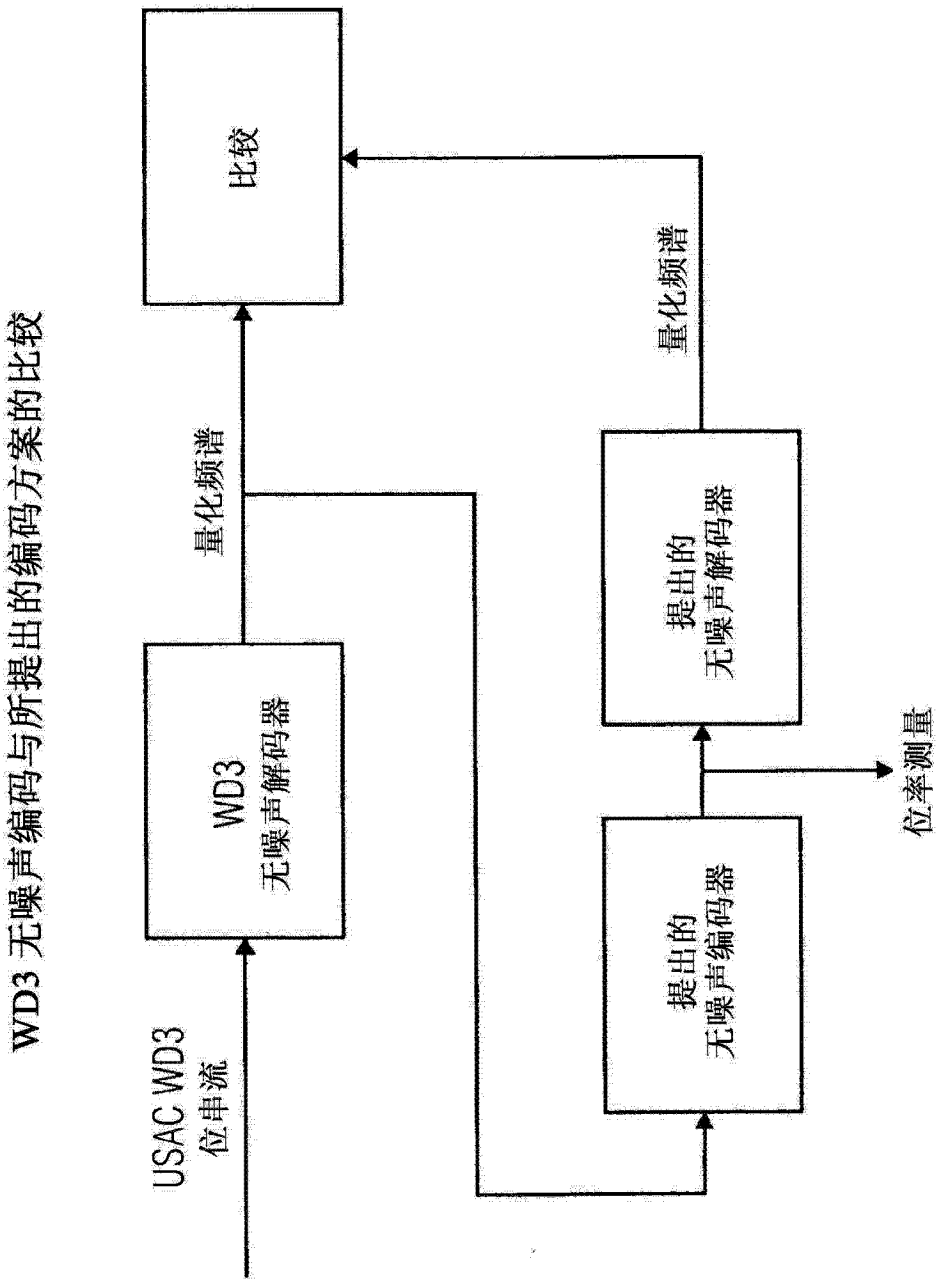

WD3 无噪声编码与所提出的编码方案的比较
USAC WD3
位串流
WD3
无噪声解码器
量化频谱
比较
提出的
无噪声编码器
提出的
无噪声解码器
量化频谱
位率测量

图 9

作为在 USAC WD4 所使用的状态计算的上下文

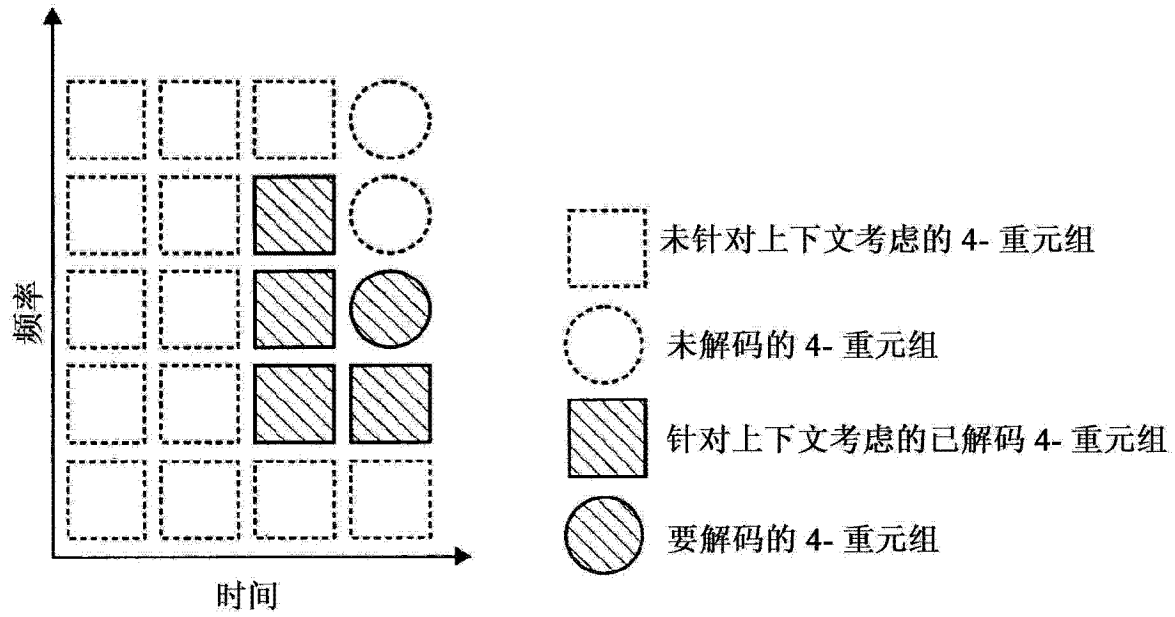


图 10A

作为在所提出的方案中所使用的状态计算的上下文

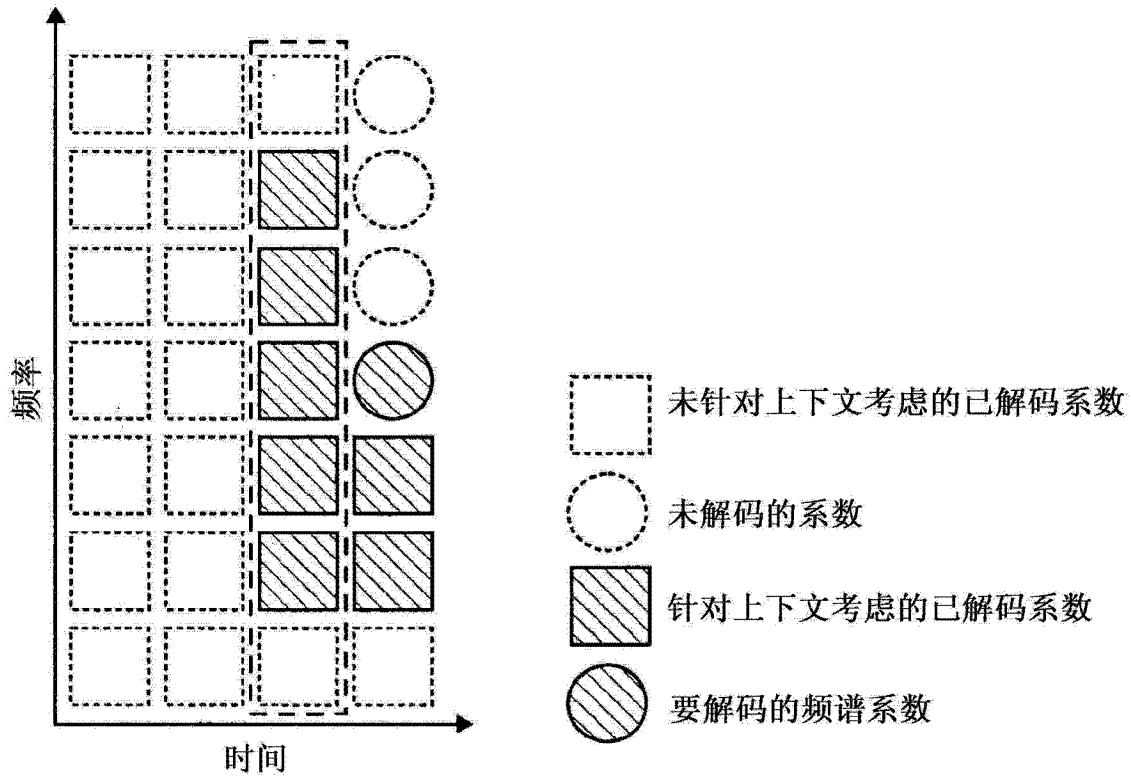


图 10B

表名	描述	数据单位	存储器（32 位的字）
arith_cf_ng_hash[128]	将上下文映射至概率模型系数的哈希表	字	128
arith_cf_ng[32][545]	针对各概率分布模 i 的累积频率组	1/2 字	8720
egroups[8][8][8][8]	4- 重元组的组系数	1/2 字	2048
dgvectors[4*4096]	组系数和元素系数至 4- 重元组的映射组	1/4 字	4096
dgroups[544]	在 dg 向量中偏移量和组中心的映射组系数	字	544
arith_cf_ne[2701]	元素系数符号的累积频率	1/2 字	1350.5
arith_cf_r[16]	较低有效位平面的累积频率	1/2 字	8
总计			**16894.5**

在 USAC WD4 算数编码方案中所使用的表

图 11A

表名	描述	数据单位	存储器（32 位的字）
arith_s_hash[387]	上下文的状态映射至累积频率表的哈希表	字	387
arith_gs_hash[225]	上下文的状态组映射至累积频率表的哈希表	字	225
arith_cf_m[64][9]	针对最高有效逐 2- 位平面 m 和 ARITH ESCAPE 符号的累积频率的模型	1/2 字	288
总计			900

在所提出的编码方案中所使用的表

图 11B

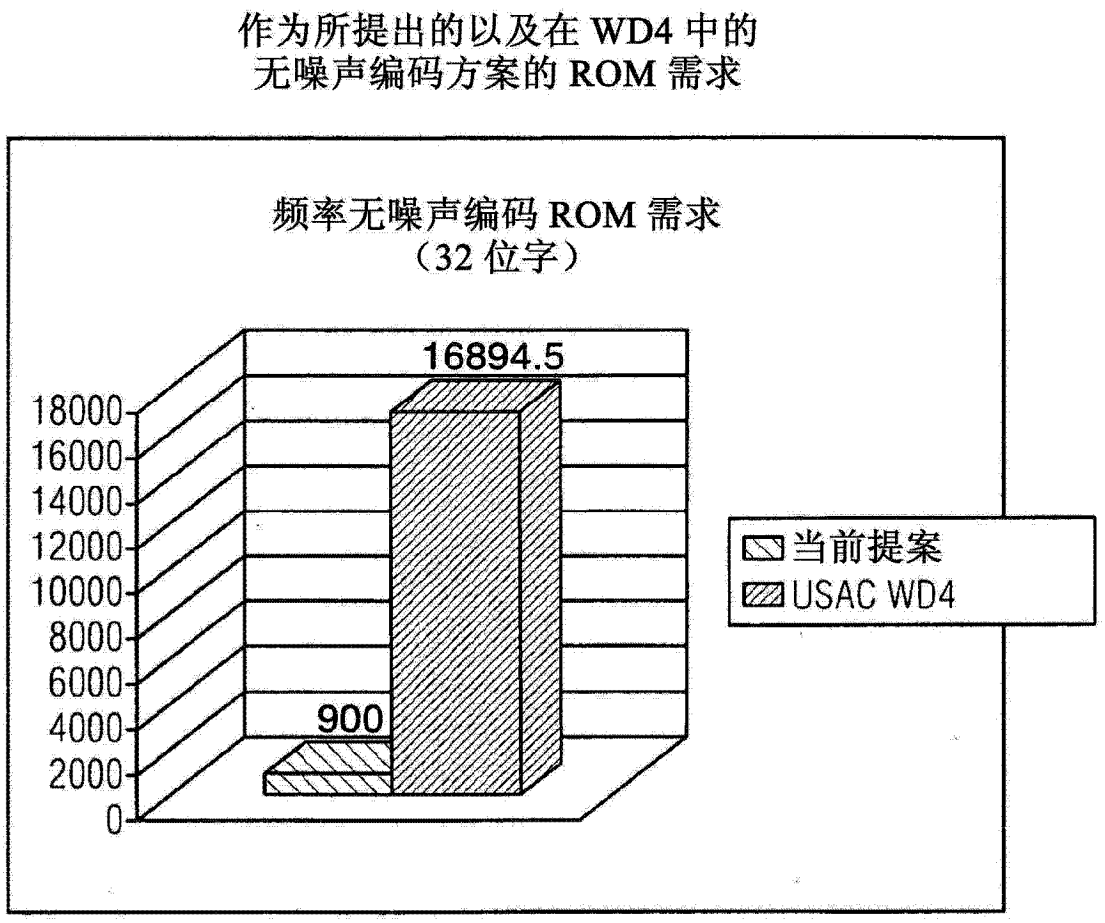

作为所提出的以及在 WD4 中的
无噪声编码方案的 ROM 需求
频率无噪声编码 ROM 需求
（32 位字）
16894.5
900
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
当前提案
USAC WD4

图 12A

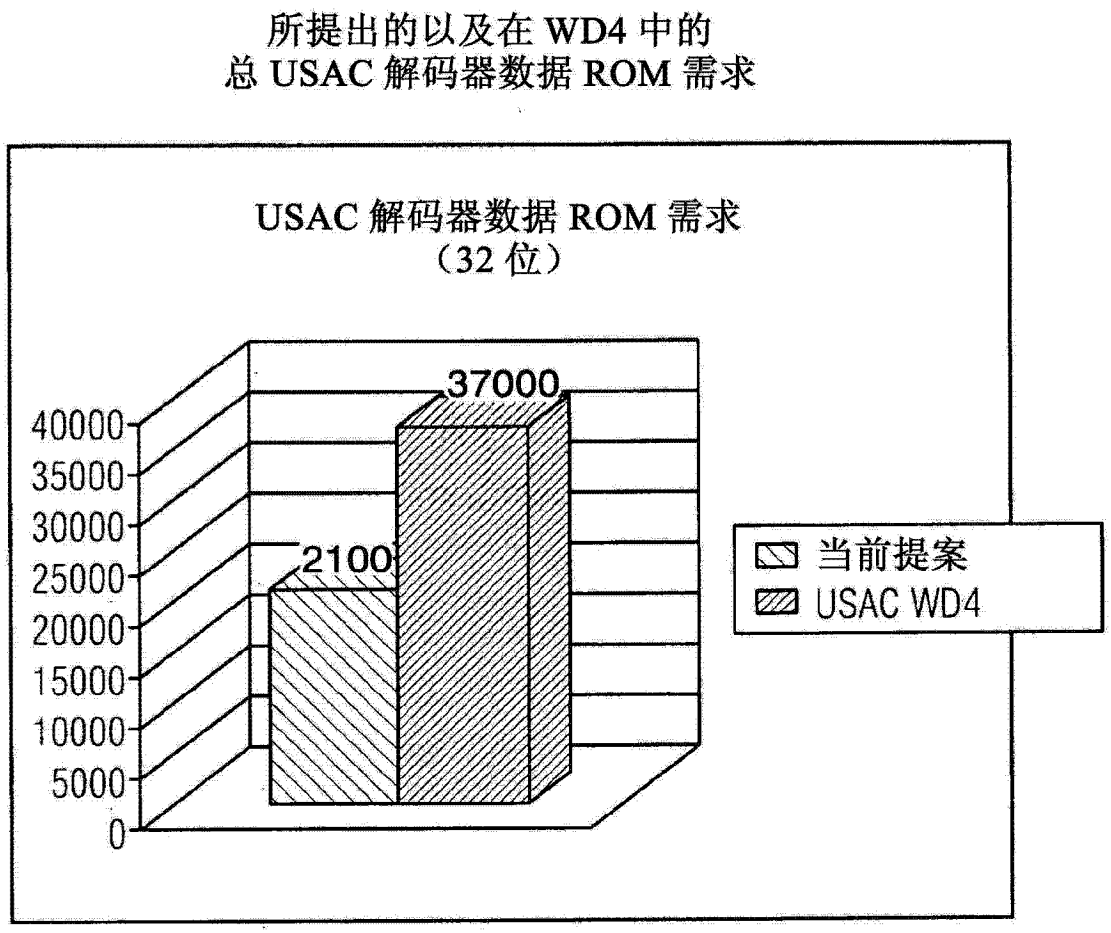

所提出的以及在 WD4 中的
总 USAC 解码器数据 ROM 需求
USAC 解码器数据 ROM 需求
（32 位）
40000
35000
30000
25000
20000
15000
10000
5000
0
37000
2100
当前提案
USAC WD4

图 12B

使用 WD 算数编码器和新提案的 USAC 编码器所产生的平均位率

操作模式	WD (kbit/s)	新提案 (kbit/s)	转码后的差异 (kbit/s)	转码后的差异（总位率 %）
Test 1, 64kbps stereo	64.00	63.34	-0.66	-1.04
Test 2, 32kbps stereo	32.00	31.66	-0.34	-1.05
Test 3, 24kbps stereo	24.00	23.73	-0.27	-1.11
Test 4, 20kbps stereo	20.00	19.78	-0.22	-1.11
Test 5, 16kbps stereo	16.00	15.82	-0.18	-1.10
Test 6, 24kbps mono	24.00	23.68	-0.32	-1.32
Test 7, 20kbps mono	20.00	19.72	-0.28	-1.39
Test 8, 16kbps mono	16.00	15.79	-0.21	-1.31
Test 9, 12kbps mono	12.00	11.86	-0.14	-1.19

图 13A

针对 USAC WD3 和新提案的位存储器控制

操作模式	位存储器控制					
	新提案			WD		
	min	max	avg	min	max	avg
Test 1, 64kbps stereo	3653	9557	8137	2314	9557	7018
Test 2, 32kbps stereo	1808	4505	4196	581	4505	3530
Test 3, 24kbps stereo	1538	4704	4408	957	4704	3871
Test 4, 20kbps stereo	2367	4864	4600	712	4864	3854
Test 5, 16kbps stereo	2712	5006	4804	724	5006	4234
Test 6, 24kbps mono	2185	4704	4457	1002	4704	3927
Test 7, 20kbps mono	2599	4864	4630	1192	4864	3935
Test 8, 16kbps mono	2820	5006	4876	1434	5006	4450
Test 9, 12kbps mono	3529	5184	5081	2256	5184	4787

图 13B

针对 USAC WD3 和新提案的平均位率

操作模式	以 kbit/s 计的平均位率					
	新提案			WD		
	FD 模式	wLPT 模式	总计	FD 模式	wLPT 模式	总计
Test 1, 64kbps stereo	53.73	---	53.73	54.40	---	54.40
Test 2, 32kbps stereo	25.31	26.34	25.60	25.80	26.61	26.02
Test 3, 24kbps stereo	18.27	19.17	18.50	18.66	19.40	18.85
Test 4, 20kbps stereo	15.50	15.93	15.61	15.83	16.12	15.90
Test 5, 16kbps stereo	12.45	12.60	12.52	12.80	12.73	12.77
Test 6, 24kbps mono	19.94	19.51	19.73	20.41	19.42	20.15
Test 7, 20kbps mono	16.15	15.91	16.08	16.56	16.12	16.45
Test 8, 16kbps mono	13.02	12.59	12.81	13.45	12.73	13.09
Test 9, 12kbps mono	9.35	9.66	9.51	9.68	9.71	9.70

图 14

以帧计的最小、最大及平均 USAC 位率

操作模式	最小位率 (kbit/s)	最大位率 (kbit/s)	平均位率 (kbit/s)
Test 1, 64kbps stereo	15.26	101.79	63.34
Test 2, 32kbps stereo	13.13	48.61	31.66
Test 3, 24kbps stereo	11.69	36.58	23.73
Test 4, 20kbps stereo	3.09	30.94	19.78
Test 5, 16kbps stereo	4.02	26.47	15.82
Test 6, 24kbps mono	1.47	37.35	23.68
Test 7, 20kbps mono	1.38	31.13	19.72
Test 8, 16kbps mono	11.40	24.64	15.79
Test 9, 12kbps mono	8.72	18.91	11.86

图 15

以帧计的最佳及最差情况

操作模式	最佳情况		最差情况	
	(bit/s)	(%)	(bit/s)	(%)
Test 1, 64kbps stereo	-30.87	-33.06	6.14	9.07
Test 2, 32kbps stereo	-10.33	-28.63	2.17	6.77
Test 3, 24kbps stereo	-11.86	-30.75	1.85	7.71
Test 4, 20kbps stereo	-7.45	-30.27	1.67	8.36
Test 5, 16kbps stereo	-5.43	-27.89	1.50	9.42
Test 6, 24kbps mono	-17.06	-45.83	1.25	4.36
Test 7, 20kbps mono	-15.86	-41.46	0.88	3.38
Test 8, 16kbps mono	-4.75	-24.85	1.11	7.31
Test 9, 12kbps mono	-3.95	-26.33	0.82	6.99

图 16

```
/*
Entropy:
fu mem.: 1.2792 bit (100.00 %)
no mem.  : 1.6289 bit (127.34 %)
split:   : 1.2971 bit (101.40 %)
*/

/*  1224 States, Entropy increase: 0.000384 */

/*Final Entropy : 1.297556 */

/*Total states = 612;*/
/*Signicant states = 387;*/
/*Pseudo states = 225;*/
/*Proba models = 64;*/
unsigned long long ari_get_pk_inc=0;
unsigned long long ari_get_pk_call_total=0;

static unsigned long ari_s_hash[387] = {
0x00000200,0x00000B01,0x00000C02,0x00000D03,0x00000F25,0x0000101C,0x0000110B,
0x00001327,
0x0000142F,0x00002B25,0x00002C22,0x00002D14,0x00002F2D,0x0000302B,0x0000312B,
0x00003330,
0x00003432,0x00003532,0x00004C32,0x00005031,0x00005131,0x0000FA02,0x0000FB01,
0x0000FC1C,
0x0000FE1C,0x0000FF1C,0x0001001E,0x00010A2E,0x00010B25,0x00010E25,0x00010F25,
0x00013938,
0x00013A04,0x00013B02,0x00013C01,0x0010393D,0x00107504,0x00107605,0x00107706,
0x0010790D,
0x00107A07,0x00107B08,0x0010850D,0x00108609,0x0010870A,0x00108B0E,0x0010B50B,
0x0010B60C,
0x0010B70D,0x0010B90B,0x0010BA1D,0x0010BB16,0x0010C615,0x0010C70C,0x0010F521,
0x0010F628,
0x0010F728,0x00110528,0x00117516,0x0011760E,0x0011770F,0x00117A12,0x00117B07,
0x0011870E,
0x0011B514,0x0011B615,0x0011B70C,0x0011B914,0x0011BA15,0x0011BB1D,0x0011C619,
0x0011C715,
0x00147516,0x00147610,0x00147711,0x0014791D,0x00147A0C,0x00147B0E,0x0014851B,
0x00148616,
0x00148707,0x0014890B,0x00148A1D,0x00148B16,0x0014950B,0x0014961D,0x0014B615,
0x0014B71D,
0x0014BA14,0x0014BB15,0x0014C614,0x0014C715,0x0015052D,0x0015750B,0x0015760C,
0x00157710,
0x0015790B,0x00157A1D,0x00157B16,0x0015850B,0x0015861D,0x0015870C,0x00158914,
0x00158A15,
0x00158B1D,0x0015B619,0x0015B714,0x0020751D,0x00207612,0x00207713,0x0020791D,
0x00207A0C,
0x00207B0E,0x0020850B,0x0020860C,0x00208710,0x00208A1D,0x00208B0C,0x0020B615,
0x0020B71D,
0x0021750B,0x0021760C,0x0021770E,0x0021790B,0x00217A1D,0x00217B12,0x00218514,
0x00218615,
0x0021870C,0x0021891C,0x00218A15,0x00218B1D,0x0021B619,0x0021B715,0x0021BA22,
0x0021BB19,
0x00247514,0x0024761D,0x0024770E,0x00247914,0x00247A15,0x00247B0C,0x00248514,
0x0024861D,
0x00248716,0x0024891C,0x00248A15,0x00248B1D,0x0024B619,0x0024B715,0x0025751C,
0x0025760B,
```

图 17	图 17(1)
	图 17(2)

图 17(1)

```
0x0025771B,0x0025791C,0x00257A0B,0x00257B1B,0x0025851C,0x0025860B,0x0025871B,
0x0025890B,
0x00258A1D,0x00258B1B,0x0025B618,0x0025B722,0x0025BA1F,0x0025BB18,0x0025C61F,
0x0025C718,
0x0025CA1F,0x0025CB1F,0x003A9D1C,0x003A9E0B,0x003A9F0B,0x004FB125,0x004FB21C,
0x004FB31C,
0x00907514,0x00907615,0x00907716,0x00907919,0x00907A15,0x00907B1D,0x00907D1C,
0x00907E1C,
0x00907F14,0x00908522,0x00908614,0x0090871D,0x00908918,0x00908A19,0x00908B14,
0x00908D24,
0x00909523,0x0090961F,0x0090992B,0x0090B517,0x0090B618,0x0090B719,0x0090B91F,
0x0090BA22,
0x0090BB19,0x0090BD1C,0x0090BE1C,0x0090BF1C,0x0090C52B,0x0090C61F,0x0090C718,
0x0090C917,
0x0090CA1F,0x0090CB1F,0x0090CD23,0x0090D52E,0x0090D62C,0x0090D92C,0x0090F52D,
0x0090F62D,
0x0090F72F,0x0090F925,0x0090FA2E,0x0090FB2D,0x0090FD1E,0x0090FE1E,0x0091052F,
0x0091062F,
0x00910928,0x00910D25,0x00917519,0x00917615,0x0091771D,0x00917922,0x00917A14,
0x00917B15,
0x00918619,0x00918714,0x00918A18,0x00918B18,0x0091B618,0x0091B722,0x0091BA18,
0x0091BB18,
0x00947518,0x00947614,0x0094771D,0x0094791F,0x00947A19,0x00947B14,0x0094852D,
0x00948619,
0x00948714,0x00948A18,0x00948B18,0x0094B61F,0x0094B718,0x0094BA17,0x0094BB1F,
0x0094C617,
0x0094C717,0x00957520,0x00957622,0x00957714,0x00957924,0x00957A18,0x00957B18,
0x00958524,
0x00958618,0x00958718,0x0095892B,0x00958A17,0x00958B1F,0x0095B52B,0x0095B617,
0x0095B71F,
0x0095B92B,0x0095BA17,0x0095BB17,0x0095C52C,0x0095C617,0x0095C717,0x0095C92C,
0x0095CA2B,
0x0095CB2C,0x00A0751F,0x00A07614,0x00A07715,0x00A0791F,0x00A07A19,0x00A07B14,
0x00A0851F,
0x00A08622,0x00A08714,0x00AC8917,0x00A08A18,0x00A08B18,0x00A0B52B,0x00A0B61F,
0x00A0B718,
0x00A0BA17,0x00A0BB1F,0x00A0C617,0x00A0C717,0x00A17524,0x00A17622,0x00A17714,
0x00A17924,
0x00A17A18,0x00A17B18,0x00A1861F,0x00A18718,0x00A18A17,0x00A18B17,0x00A1B617,
0x00A1B71F,
0x00A1BA17,0x00A1BB17,0x00A47524,0x00A47618,0x00A47714,0x00A4792B,0x00A47A18,
0x00A47B18,
0x00A4852B,0x00A4861F,0x00A48718,0x00A4892B,0x00A48A17,0x00A48B17,0x00A4B52C,
0x00A4B617,
0x00A4B717,0x00A4B92C,0x00A4BA17,0x00A4BB17,0x00A4C52E,0x00A4C62B,0x00A4C72B,
0x00A4C92E,
0x00A4CA2C,0x00A4CB2C,0x00A5752C,0x00A57617,0x00A57718,0x00A5792B,0x00A57A17,
0x00A57B1F,
0x00A5852C,0x00A58617,0x00A5871F,0x00A5892C,0x00A58A17,0x00A58B17,0x00A5B52E,
0x00A5B62B,
0x00A5B717,0x00A5B92E,0x00A5BA2C,0x00A5BB2C,0x00A5C52E,0x00A5C62C,0x00A5C72C,
0x00A5C92E,
0x00A5CA2C,0x00A5CB2C,0x00BA9D2D,0x00BA9E2E,0x00BA9F2E,0x00CDD938,0x00CE2D38,
0x00CEE938,
0x00CF2D38,0x00D02D38,0x00D0313A,0x00D0713A,0x0110762F,0x0110B632,0x0110B731,
0x03D0113B,
0x03D0213C,0x03D02D3D,0x03D0313D,0x03D0413C,0x03D04D3D,0x03D0513C,0x03D05D3D,
0x03D06139,
0x03D0693D,0x03D06D3D,0x03D0713D
};
```

图 17	图 17(1)
	图 17(2)

图 17(2)

```
static unsigned long ari_gs_hash[225] = {
0x00000401,0x0001491A,0x0001590B,0x00017621,0x0001891C,0x0009492A,0x000DFA38,
0x000F6D3D,
0x0010003B,0x0010B21B,0x0011321C,0x00116B29,0x0011B31D,0x00126B1E,0x00136623,
0x00146729,
0x00146F3B,0x0015321F,0x00156E27,0x00163320,0x00182725,0x00186727,0x00196323,
0x001C4721,
0x001E3F30,0x001E433B,0x00203F2A,0x0020463B,0x0020F322,0x00216A2E,0x00226723,
0x00245625,
0x00256724,0x00286625,0x002D3726,0x002D573A,0x00316627,0x00326628,0x00344729,
0x00366628,
0x003D4329,0x00416A2A,0x0042533A,0x00916A2A,0x00926B2B,0x0093E72E,0x00956B2C,
0x009D362D,
0x009D3B39,0x009E4330,0x00A2672E,0x00AD372F,0x01145630,0x01146B27,0x011C8231,
0x01226732,
0x012CC333,0x01413B34,0x019CA335,0x019CB338,0x01ACB736,0x01AD823D,0x01C37F37,
0x02156738,
0x0218AB3B,0x021C9B35,0x021E0738,0x021FB73D,0x0220E335,0x02216B3C,0x02217234,
0x0222B33C,
0x02239B3B,0x0223B23A,0x0224673B,0x0238A739,0x0240B23D,0x024CBF38,0x024CC23D,
0x024D8738,
0x0297AF3A,0x02986727,0x0298A33B,0x0298A738,0x029CAF3B,0x029CC33A,0x02A0AB35,
0x02A3E736,
0x02AC773B,0x02B0B335,0x02B3A73B,0x02C0D73C,0x02C1E735,0x03108E3D,0x03109737,
0x0311D639,
0x03147F3C,0x0314B236,0x0317A639,0x0317D629,0x0317DB33,0x03187627,0x0318AF3B,
0x0318F61A,
0x0319D739,0x031C953B,0x031D633C,0x031FCF39,0x0320873B,0x0320963A,0x03222639,
0x0323833C,
0x03239A27,0x0323EA2F,0x03242631,0x03242B3B,0x03249727,0x0325AB39,0x0327A73C,
0x0327C728,
0x03287727,0x03287E3A,0x03288737,0x032BAA39,0x032C7527,0x032D2337,0x032E9B39,
0x032EA23B,
0x032EBF3C,0x032F7E39,0x0330C63C,0x0332B23B,0x0332F230,0x03339F3B,0x0333EE27,
0x03348F30,
0x0336AB3C,0x0338A73B,0x033A7639,0x033A7F1A,0x033C793B,0x033C9A34,0x033CA33B,
0x033CA738,
0x033D0A3C,0x033DB339,0x033DFF3C,0x033E9739,0x0340CB3C,0x0344573B,0x0344AA3C,
0x0348263B,
0x034C7B3C,0x034CBB3A,0x034CD33C,0x0390B73D,0x0390E937,0x0393653D,0x0394B73B,
0x0394E33D,
0x0394FA38,0x03950A3C,0x0396CF3D,0x03971A36,0x0398673C,0x0398E13B,0x03994E39,
0x039C733B,
0x039D191A,0x039D4536,0x039E053C,0x039E6E3D,0x039E9D34,0x039F8D39,0x03A0C93B,
0x03A67939,
0x03A69D29,0x03A6D637,0x03A85A3C,0x03AE5B3B,0x03AEDB3D,0x03AF2E3C,0x03B0A13B,
0x03B2B139,
0x03B3123B,0x03B36339,0x03B3AD3C,0x03B42E33,0x03B4733B,0x03B4F53C,0x03B51F36,
0x03B59139,
0x03B5CB3C,0x03B61737,0x03B93A3C,0x03B98F39,0x03B9F53C,0x03BA063B,0x03BA2A3C,
0x03BB2739,
0x03BD3B3B,0x03BDC939,0x03BDF534,0x03BF9A39,0x03C1653B,0x03C19E2A,0x03C20527,
0x03C3633B,
0x03C3823C,0x03C3A527,0x03C45A3B,0x03C4993C,0x03C5B23B,0x03C5D527,0x03C9563B,
0x03C9A93C,
0x03CA063B,0x03CB0E3C,0x03CCB53B,0x03CD1E3C,0x03CED23D,0x03CEDF3C,0x03CFFA39,
0x40BC673E,
0xFFFFFF3F
};
```

图 18

```
unsigned short ari_cf_m[64][9] = {
{65535,65534,65532,65215,  321,    4,    2,    1,    0},
{65490,65339,64638,58133, 7463,  973,  270,  125,    0},
{65530,65509,65319,60216, 5308,  222,   30,    9,    0},
{65534,65528,65470,62535, 3012,   67,    8,    2,    0},
{65533,65524,65435,62110, 3434,  104,   14,    5,    0},
{65535,65533,65499,62363, 3173,   37,    3,    1,    0},
{65535,65534,65522,63164, 2371,   14,    2,    1,    0},
{65535,65530,65448,59939, 5612,   88,    7,    2,    0},
{65535,65533,65500,61498, 4044,   38,    3,    1,    0},
{65535,65530,65444,59855, 5667,   92,    6,    1,    0},
{65535,65532,65495,61386, 4140,   39,    3,    1,    0},
{65522,65458,64905,55424,10056,  634,   88,   28,    0},
{65532,65511,65238,57072, 8457,  297,   27,    6,    0},
{65534,65522,65364,59096, 6461,  171,   15,    3,    0},
{65535,65530,65426,59204, 6342,  109,    8,    2,    0},
{65535,65533,65492,61008, 4512,   43,    3,    1,    0},
{65535,65529,65417,58998, 6519,  118,    6,    1,    0},
{65535,65533,65490,60856, 4679,   46,    4,    1,    0},
{65535,65528,65384,58400, 7127,  149,    9,    1,    0},
{65535,65532,65483,60544, 4984,   56,    4,    1,    0},
{65517,65413,64537,53269,12264, 1002,  138,   38,    0},
{65531,65503,65125,55553, 9985,  420,   37,    7,    0},
{65534,65518,65303,57889, 7650,  235,   20,    3,    0},
{65490,65288,63679,49500,15949, 1903,  301,   94,    0},
{65522,65428,64429,51580,13957, 1113,  114,   22,    0},
{65526,65447,64600,52808,12743,  937,   93,   17,    0},
{63814,60228,53108,40709,26294,15412, 8961, 5729,    0},
{65526,65486,65133,57227, 8244,  400,   58,   20,    0},
{65500,65346,64297,52845,12477, 1283,  230,   70,    0},
{65528,65486,65077,56652, 8871,  465,   56,   16,    0},
{65464,65186,63581,50731,14351, 1992,  396,  128,    0},
{65489,65278,63861,51225,14185, 1726,  302,   96,    0},
{65485,65249,63632,50425,14933, 1943,  332,   96,    0},
{65292,64495,61270,47805,17600, 4502, 1337,  542,    0},
{65519,65421,64478,52517,12971, 1068,  129,   33,    0},
{65470,65181,63344,49862,15299, 2233,  418,  132,    0},
{65472,65197,63407,49933,15445, 2176,  396,  123,    0},
```

1910 → 第一行

1912 → 第二行

第一行 ← pki=0

第二行 ← pki=1

图 19	图 19(1)
	图 19(2)

图 19(1)

```
{65376,64781,62057,48496,16676, 3614,  923,  340,    0},
{65259,64356,60836,47316,18158, 4979, 1517,  623,    0},
{64883,63190,58260,45006,21034, 8378, 3559, 1909,    0},
{65261,64180,60126,46710,18694, 5578, 1582,  531,    0},
{64933,63355,58991,46299,19470, 7245, 2989, 1449,    0},
{63999,61383,56309,44712,24964,14237, 9489, 7028,    0},
{65451,65091,62953,48747,16324, 2626,  522,  168,    0},
{65400,64870,62109,47037,18198, 3526,  794,  278,    0},
{65200,64074,59673,44322,20692, 6133, 1836,  739,    0},
{65376,64798,61822,46437,18673, 3881,  932,  368,    0},
{65151,63887,59083,43617,21491, 6768, 2081,  841,    0},
{64592,62314,56211,42184,24450,11142, 5265, 3075,    0},
{64908,62840,56205,41474,23652, 9844, 3388, 1379,    0},
{65021,63308,57341,42286,22972, 8709, 2895, 1232,    0},
{64790,62474,55461,40843,24327,10719, 3921, 1677,    0},
{64053,60476,52429,39583,26962,15208, 7592, 4166,    0},
{63317,58934,51305,40469,29263,19682,12661, 8553,    0},
{63871,59872,52031,39473,26093,15132, 7866, 4080,    0},
{63226,58553,50425,39191,28586,18779,11388, 7035,    0},
{62219,57006,49569,40492,32376,24784,18716,14447,    0},
{62905,58273,50651,39619,28123,18379,11633, 7478,    0},
{63420,59073,51922,41516,29863,20328,13529, 9237,    0},
{63582,59263,51165,37880,24026,13893, 7771, 4535,    0},
{63223,58418,49833,37279,25503,15421, 9122, 5802,    0},
{62322,56878,48746,39095,30723,22195,15849,11887,    0},
{61826,47222,47123,47015,46913,46806,13713, 6895,    0},
{60678,44085,44084,44083,44082,44081,16715, 9222,    0}
};
```

1964 → 指向倒数第二行

← pki=63 指向最后一行

图 19	图 19(1)
	图 19(2)

图 19(2)

```
static unsigned long ari_s_hash[387] = {
0x0090D52E,0x0090CB1F,0x00A4CB2C,0x00003330,0x00107A07,0x00907A15,0x00207A0C,
0x00147A0C,
0x00247A15,0x00A07A19,0x00947A19,0x00A47A18,0x0010B70D,0x0090B719,0x0020B71D,
0x0014B71D,
0x00A0B718,0x0094B718,0x0024B715,0x00A4B717,0x0110B731,0x0000FE1C,0x0090FE1E,
0x00013B02,
0x00A5C92E,0x0095C92C,0x003A9E0B,0x00000B01,0x00BA9E2E,0x0090992B,0x0011B514,
0x00A5B52E,
0x0095B52B,0x0090D62C,0x0010850D,0x0014851B,0x00248514,0x0020850B,0x00908522,
0x00948520,
0x00A4852B,0x00A0851F,0x00003432,0x00107B08,0x00207B0E,0x00907B1D,0x00147B0E,
0x00247B0C,
0x00A07B14,0x00947B14,0x00A47B18,0x00910928,0x03D0713D,0x00D0713A,0x0000FF1C,
0x0090F52D,
0x0010F521,0x00013C01,0x03D05D3D,0x00A5CA2C,0x0025CA1F,0x0095CA2B,0x003A9F0B,
0x00000C02,
0x0021790B,0x0025791C,0x00917922,0x0015790B,0x00A17924,0x00A5792B,0x00957924,
0x00BA9F2E,
0x00CF2D38,0x00000200,0x0011B615,0x0091B618,0x0021B619,0x0015B619,0x00A1B617,
0x0095B617,
0x0025B618,0x00A5B62B,0x004FB125,0x00108609,0x00148616,0x0020860C,0x00908614,
0x0024861D,
0x00948619,0x00A08622,0x00A4861F,0x0090CD23,0x00003532,0x00010A2E,0x00002B25,
0x0010B90B,
0x0090B91F,0x00A4B92C,0x0001001E,0x03D0213C,0x0090F62D,0x0010F628,0x00A5CB2C,
0x0025CB1F,
0x0095CB2C,0x00000D03,0x00117A12,0x00217A1D,0x00917A14,0x00257A0B,0x00157A1D,
0x00A17A18,
0x00A57A17,0x00957A18,0x0011B70C,0x0091B722,0x0021B715,0x0015B714,0x00A1B71F,
0x0025B722,
0x0095B71F,0x00A5B717,0x004FB21C,0x0010870A,0x00148707,0x00208710,0x0090871D,
0x00248716,
0x00948714,0x00A08714,0x00A48718,0x00907D1C,0x00010B25,0x00002C22,0x0010BA1D,
0x0090BA22,
0x0014BA14,0x00A0BA17,0x0094BA17,0x00A4BA17,0x03D0693D,0x0090F72F,0x0010F728,
0x0025851C,
0x0015850B,0x00218514,0x00A5852C,0x00958524,0x00117B07,0x00217B12,0x00257B1B,
0x00917B15,
0x00157B16,0x00A17B18,0x00A57B1F,0x00957B18,0x0090D92C,0x004FB31C,0x03D0413C,
0x00907E1C,
0x0090C52B,0x00A4C52E,0x00002D14,0x00D02D38,0x03D02D3D,0x0010BB16,0x0090BB19,
0x0014BB15,
0x00A0BB1F,0x0094BB1F,0x00A4BB17,0x0025860B,0x0015861D,0x00218615,0x00918619,
0x00A58617,
0x00958618,0x00A1861F,0x00000F25,0x00CEE938,0x00004C32,0x0011B914,0x00A5B92E,
0x0095B92B,
0x0014890B,0x0024891C,0x00908918,0x00A4892B,0x00A08917,0x00907F14,0x0010C615,
0x0090C61F,
0x0014C614,0x0094C617,0x00A4C62B,0x00A0C617,0x00910D25,0x00107504,0x00907514,
0x00147516,
0x0020751D,0x00247514,0x00A0751F,0x00947518,0x00A47524,0x0090F925,0x0011870E,
0x0025871B,
0x0015870C,0x0021870C,0x00918714,0x00A5871F,0x00A18718,0x00958718,0x03D06139,
0x0000101C,
0x03D04D3D,0x0011BA15,0x0091BA18,0x0021BA22,0x00A1BA17,0x0025BA1F,0x00A5BA2C,
0x0095BA17,
0x00148A1D,0x00208A1D,0x00248A15,0x00908A19,0x00948A18,0x00A08A18,0x00A48A17,
0x0010393D,
```

图 20	图 20(1)
	图 20(2)

图 20(1)

```
0x0010C70C,0x0090C718,0x0014C715,0x0094C717,0x00A0C717,0x00A4C72B,0x00010E25,
0x00002F2D,
0x00107605,0x00907615,0x00147610,0x00207612,0x0024761D,0x00A07614,0x00947614,
0x00A47618,
0x0110762F,0x0090BD1C,0x00CDD938,0x0000FA02,0x0090FA2E,0x0000110B,0x03D0113B,
0x00A5C52E,
0x0095C52C,0x0011BB1D,0x0091BB18,0x0021BB19,0x0014950B,0x00A1BB17,0x0025BB18,
0x00A5BB2C,
0x0095BB17,0x00909523,0x00108B0E,0x00148B16,0x00208B0C,0x00248B1D,0x00908B14,
0x00948B18,
0x00A08B18,0x00A48B17,0x00010F25,0x0000302B,0x00107706,0x00907716,0x00147711,
0x00207713,
0x0024770E,0x00A07715,0x0094771D,0x00A47714,0x0091052F,0x00110528,0x0090BE1C,
0x0015052D,
0x03D06D3D,0x0000FB01,0x0090FB2D,0x0025890B,0x00A5892C,0x00158914,0x0021891C,
0x0095892B,
0x0011C619,0x0025C61F,0x00A5C62C,0x0095C617,0x00117516,0x0021750B,0x0015750B,
0x00917519,
0x0025751C,0x00A17524,0x00957520,0x00A5752C,0x0014961D,0x0090961F,0x0090C917,
0x00A4C92E,
0x0000312B,0x00D0313A,0x03D0313D,0x0090BF1C,0x0091062F,0x0010B50B,0x0090B517,
0x00A0B52B,
0x00A4B52C,0x0000FC1C,0x00258A1D,0x00A58A17,0x00158A15,0x00218A15,0x00918A18,
0x00A18A17,
0x00958A17,0x00013938,0x00001327,0x0011C715,0x0025C718,0x00A5C72C,0x0095C717,
0x0011760E,
0x0021760C,0x00917615,0x0015760C,0x0025760B,0x00A17622,0x00957622,0x00A57617,
0x00005031,
0x00908D24,0x0090CA1F,0x00A4CA2C,0x0010790D,0x00907919,0x0020791D,0x0014791D,
0x00247914,
0x00A0791F,0x0094791F,0x00A4792B,0x00CE2D38,0x0010B60C,0x0090B618,0x0014B615,
0x0020B615,
0x00A0B61F,0x0094B61F,0x0024B619,0x00A4B617,0x0110B632,0x00258B1B,0x00158B1D,
0x00218B1D,
0x00A58B17,0x00918B18,0x00A18B17,0x00958B1F,0x0090FD1E,0x00013A04,0x0000142F,
0x003A9D1C,
0x00BA9D2D,0x0011770F,0x0021770E,0x00157710,0x0091771D,0x0025771B,0x00A17714,
0x00957714,
0x00A57718,0x00005131,0x03D0513C
};
```

图 20	图 20(1)
	图 20(2)

图 20(2)

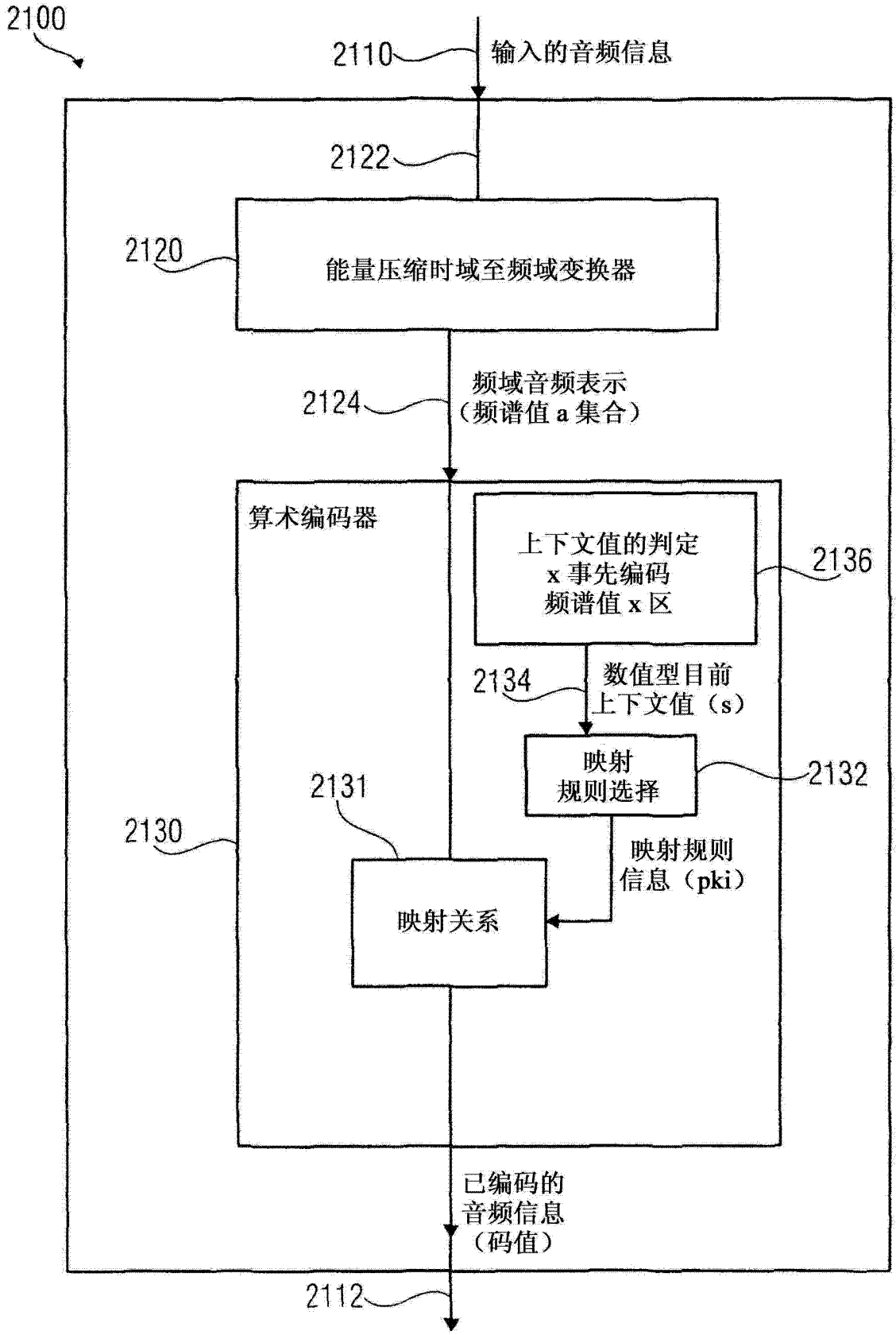

2100
2110
输入的音频信息
2122
2120
能量压缩时域至频域变换器
2124
频域音频表示
（频谱值 a 集合）
算术编码器
上下文值的判定
x 事先编码
频谱值 x 区
2136
2134
数值型目前
上下文值（s）
映射
规则选择
2132
2131
2130
映射规则
信息（pki）
映射关系
已编码的
音频信息
（码值）
2112

图 21

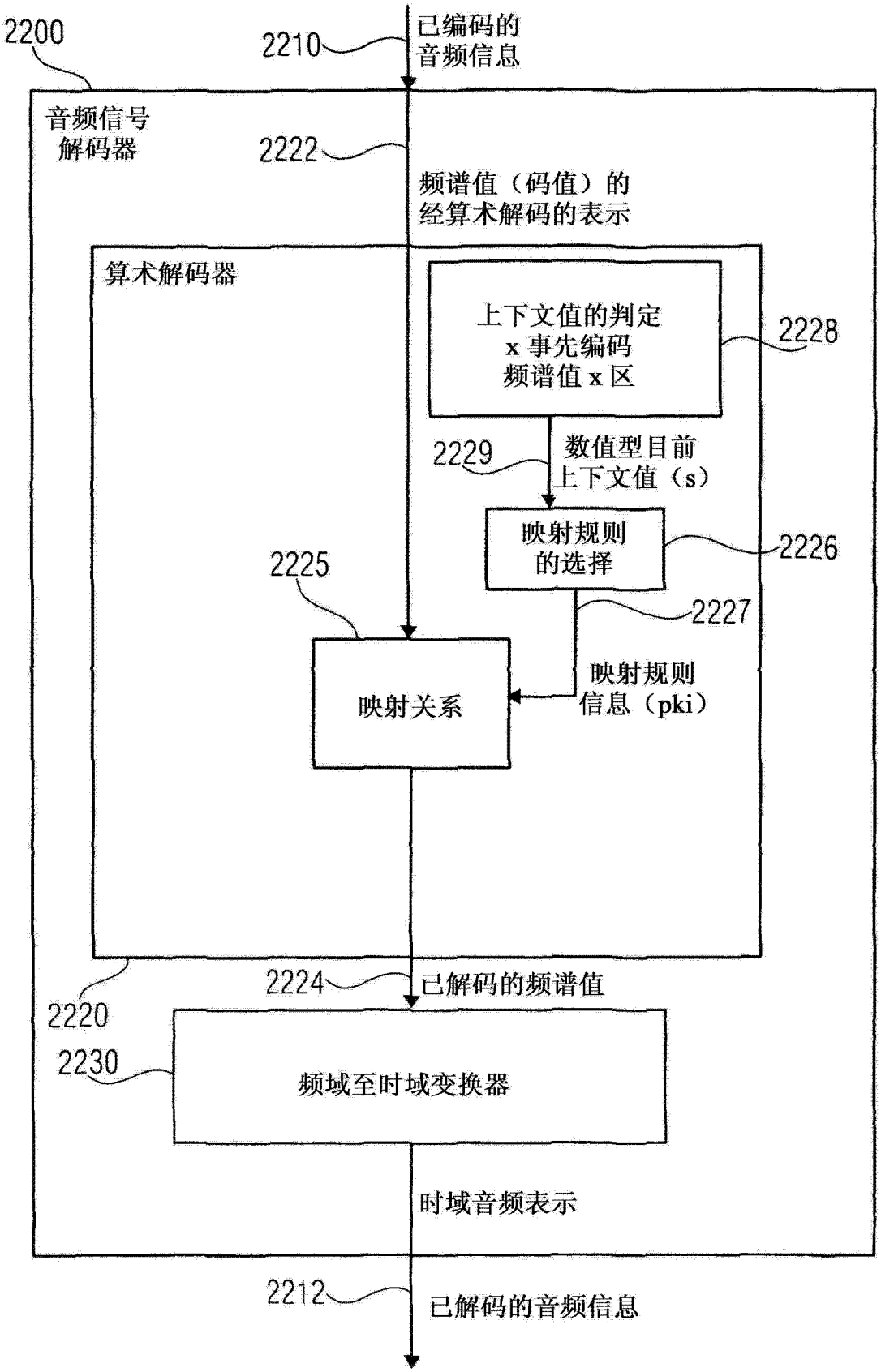

2200
2210
已编码的
音频信息
音频信号
解码器
2222
频谱值（码值）的
经算术解码的表示
算术解码器
上下文值的判定
x 事先编码
频谱值 x 区
2228
2229
数值型目前
上下文值（s）
映射规则
的选择
2226
2225
2227
映射关系
映射规则
信息（pki）
2224
已解码的频谱值
2220
2230
频域至时域变换器
时域音频表示
2212
已解码的音频信息

图 22