[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H03M 13/29 (2006.01)
G11B 20/18 (2006.01)

[12] 发明专利说明书

专利号 ZL 03807686.1

[45] 授权公告日 2009 年 9 月 9 日

[11] 授权公告号 CN 100539445C

[22] 申请日 2003.3.14 [21] 申请号 03807686.1

[30] 优先权

[32] 2002.4.5 [33] EP [31] 02076354.6

[86] 国际申请 PCT/IB2003/001068 2003.3.14

[87] 国际公布 WO2003/085840 英 2003.10.16

[85] 进入国家阶段日期 2004.9.30

[73] 专利权人 皇家飞利浦电子股份有限公司

地址 荷兰艾恩德霍芬

共同专利权人 索尼株式会社

[72] 发明人 M·E·范迪克 K·亚马莫托

M·哈特托里

[56] 参考文献

EP0603932A1 1994.6.29

WO0007300A1 2000.2.10

US5299208A 1994.3.29

Channel coding and signal processing for optical recordingsystems beyond DVD. COENE W ET AL. IEEE TRANSACTIONS ONMAGNETICS, Vol. 37 No. 2. 2000

On performance of product codes and comparison withinterleaved BCH codes in different mobile channels. YUAN D F ET AL. INTERNATIONAL CONFERENCE ON GLOBAL CONNECTIVITY IN ENERGY, COMPUTER, COMMUNICATION AND CONTROL. 1998

审查员 程 琼

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 李亚非 王忠忠

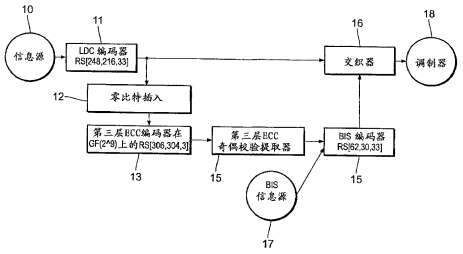
权利要求书 2 页 说明书 9 页 附图 7 页

[54] 发明名称

将纠错附加层嵌入纠错码的方法和装置

[57] 摘要

本发明涉及将纠错附加层嵌入纠错码的方法，其中信息被编码到第一伽罗瓦域上的代码的码字中，其中多个码字被排列在包括用户数据子块和奇偶校验数据子块的编码块列中。为了提供易于实现、保持兼容性并能提高纠错能力的纠错附加层，提出的方法包括以下步骤：－利用比第一伽罗瓦域大的第二伽罗瓦域上的水平纠错码独立或成组地对用户数据子块的行编码，以便获得水平奇偶校验，－将水平奇偶校验作为附加层嵌入纠错码。



1. 将纠错附加层嵌入纠错码的方法，其中用户信息被编码为在第一伽罗瓦域上的代码的码字，其中多个码字排列在包含用户数据子块和奇偶校验数据子块的代码块列中，该方法包括以下步骤：

- 利用比所述第一伽罗瓦域大的第二伽罗瓦域上的水平纠错码独立或成组地对至少所述用户数据子块的行编码，以便获得水平奇偶校验，

- 将所述水平奇偶校验作为附加层嵌入纠错码。

2. 根据权利要求 1 的方法，其中在对所述用户数据子块行编码之前，将具有预定义值的预定义数目的比特添加到所述用户数据子块的每个符号中。

3. 根据权利要求 2 的方法，其中将具有零比特值的一个或两个比特添加到所述用户数据子块的每个符号中。

4. 根据权利要求 1 的方法，其中所述代码块是包含 LDC 码字的长距码 LDC 块，并且所述 LDC 码字排列在 LDC 块的列中。

5. 根据权利要求 4 的方法，其中所述 LDC 码字是第一伽罗瓦域 GF（2^8）上的码字。

6. 根据权利要求 4 的方法，其中利用伽罗瓦域 GF(2^9)上的[306，304，3]Reed Solomon 代码独立地对用户数据子块的每行编码。

7. 根据权利要求 4 的方法，其中利用 Reed Solomon 码的子空间子码，独立地对所述用户数据子块的每行编码。

8. 根据权利要求 7 的方法，其中利用伽罗瓦域（2^9）上的 Reed Solomon 码的子空间子码，独立地对所述用户数据子块的每行编码。

9. 根据权利要求 4 的方法，其中利用伽罗瓦域 GF（2^{10}）上的 Reed Solomon 代码，以至少两个连续行为一组，成组地对用户数据子块的行编码。

10. 根据权利要求 9 的方法，其中所述至少两个连续行是三个连续行。

11. 根据权利要求 4 的方法，其中利用 Reed Solomon 码的子空间子码，以至少两个连续行为一组，成组地对所述用户数据子块的行编码，成组地对用户数据子块的行编码。

12. 根据权利要求 11 的方法，其中利用伽罗瓦域 GF（2^{10}）上的 Reed Solomon 码的子空间子码，以三个连续行为一组，成组地对用户数据子块的行编码。

13. 根据权利要求 1 的方法，其中独立或成组地对完整代码块的行编码。

14. 根据权利要求 1 的方法，其中利用附加纠错码，对水平奇偶校验编码。

15. 根据权利要求 14 的方法，所述附加纠错码是包含 GF（2^8）上的 Reed Solomon 代码的脉冲串指示子码 BIS。

16. 对根据权利要求 1 的方法，在其中嵌入了纠错附加层的纠错码解码的方法，其中用户信息被编码为第一伽罗瓦域上的代码的码字，其中多个码字排列在包含用户数据子块和奇偶校验数据子块的代码块列中，该方法包括以下步骤：

- 从纠错码中提取所述水平奇偶校验，

- 利用比使用水平奇偶校验的第一伽罗瓦域大的第二伽罗瓦域上的、已经在权利要求 1 的方法中用于编码的水平纠错码独立或成组地对至少用户数据子块的行解码。

17. 将纠错附加层嵌入纠错码的装置，其中用户信息被编码为第一伽罗瓦域上的代码的码字，其中多个码字排列在包含用户数据子块和奇偶校验数据子块的代码块的列中，该装置包括：

- 利用比第一伽罗瓦域大的第二伽罗瓦域上的水平纠错码独立或成组地对至少所述用户数据子块的行编码，以便获得水平奇偶校验的装置，

将水平奇偶校验作为附加层嵌入纠错码的装置。

18. 对根据权利要求 1 的方法在其中嵌入了纠错附加层的纠错码解码的装置，其中用户信息被编码为第一伽罗瓦域上的代码的码字，其中多个码字排列在包含用户数据子块和奇偶校验子块的代码块列中，该装置包括：

- 从纠错码中提取水平奇偶校验的装置，

- 利用比使用水平奇偶校验的第一伽罗瓦域大的第二伽罗瓦域上的、已经在权利要求 1 的方法中用于编码的水平纠错码独立或成组地对至少用户数据子块的行解码的装置。

将纠错附加层嵌入纠错码的方法和装置

技术领域

本发明涉及将纠错附加层嵌入纠错码的方法，其中信息被编码为第一伽罗瓦域上的代码的码字，其中多个码字被排列在包括用户数据子块和奇偶校验数据子块的编码块列中。本发明还涉及对这种纠错码解码的方法、以及相应的装置、存储这种编码的码字的存储媒质、包含这种码字的信号和实现该方法的计算机程序。

背景技术

利用分字交错(wordwise interleaving)对多字信息编码的方法公开在WO 00/07300中。在那里，描述了一种包含两种类型的码字，长程码 LDC（Long Distance Code）和突发指示子码 BIS（Burst Indicator Subcode），的所谓的哨兵码（picket code），该编码已被确定用于DVR(数字视频记录)，以便在光记录载体上存储数据，特别是视频数据。BIS码字提供较强的纠错能力。即使在最恶劣的情况下，不能对其正确解码几乎不可能。在对BIS列解码之后，可以将突发错误识别出来。在应用擦除策略之后，也能对纠错能力较弱的LDC码字正确解码。

发明内容

与已有的纠错码相比，例如DVD中的乘积码（product code），哨兵码提高了纠正（多个）突发错误的能力。然而，哨兵码纠正随机错误的能力较弱。因此，本发明的目的是提供一种提高纠错能力，特别是纠正随机错误的方法，该方法易于实现且兼容现有纠错码方案。所述的方法可以应用于任何纠错码，特别是用于DVR的哨兵码。

该目的是本发明的将纠错附加层嵌入纠错码的方法实现的，包括以下步骤：

- 利用水平纠错码在大于第一伽罗瓦域的第二伽罗瓦域上独立或成组地对用户数据块的行编码，以便获得水平奇偶校验，
- 将水平奇偶校验作为附加层嵌入纠错码。

本发明基于常规概念，即对排列在代码块列中的码字产生水平奇偶校验。因为代码块的域大小，即代码块的每行的长度，通常大于代

码块列中的码字所归属的第一伽罗瓦域上的最长代码，所以使用更大伽罗瓦域上的代码。更大伽罗瓦域的大小应该使得较大伽罗瓦域上的代码的码字长度大于需要产生附加水平奇偶校验的代码块的行的长度。然后，所产生的水平奇偶校验作为附加层嵌入原始代码，用于纠错。该附加层在解码过程中使用，用于纠正擦除、突发错误和解码故障。因为所获得的水平奇偶校验受到额外的保护，所以根据本发明可以获得高水平的纠错能力。

本发明还涉及对根据上述编码方法将纠错附加层嵌入其中的纠错码解码的方法，该方法包括以下步骤：

- 从纠错码中提取水平奇偶校验，

- 利用已经在权利要求 1 所述的方法中用于编码的水平纠错码，在比使用水平奇偶校验的第一伽罗瓦域大的第二伽罗瓦域上独立或成组地对至少用户数据字块的行解码。

此外，本发明还涉及将纠错附加层嵌入纠错码的装置以及相应的解码装置，以及按照将纠错附加层嵌入其中的纠错码的码字形式存储数据的存储媒质，特别是光记录载体例如 CD、DVD 或 DVR 盘，包含码字形式的数据的信号，和包含当程序在计算机上运行时能够使计算机相应编解码的方法的程序代码方法的计算机程序。本发明的优选实施方案定义在附属权利要求中。

为了在较大的伽罗瓦域上对行编码，在编码前，将具有预定义值的预定义数目的比特添加到用户数据子块的每个符号上。最容易的方法是向每个符号添加一个零比特值的比特。然而，一般情况下，任何其它比特值都是可以的。还可能的是，将具有任何比特值的比特添加到每个符号，即不是所有的比特都需要具有相同的比特值。然而，有必要将添加的比特序列存储在一行的符号中，因为必须将相同的序列添加到每一行。在这种情况下，需要存储该序列的寄存器。然而，当添加零比特值的比特时，不需要用于存储的寄存器；此外，执行的“与”操作也较少。

优选的是，本发明应用于使用如 DVR 所采用的哨兵码的方法，其中代码块是包含排列在 LDC 块列中的 LDC 码字，特别是 GF（2^8）上的[248，216，33]码字，的 LDC 块，其中还使用了 BIS 码字，特别是 GF（2^8）上的[62，30，33]Reed Solomon 码字。优选的是，获得的

水平奇偶校验由附加的纠错码编码，即当使用哨兵码 picket code 时，优选地对水平奇偶校验编码并写入 BIS 码字。

当将本发明应用于哨兵码 picket code 时，用户数据子块的每行都优选地利用伽罗瓦域 GF（2^9）上的 [306，304，3] Reed Solomon（RS）代码编码。这意味着将一个额外的比特添加到用户数据子块的每个符号上，所以每个符号包含 9 个比特。如上所述，优选的是将一个零比特值的比特添加到每个符号上。利用 GF（2^9）上的代码，可以获得两个额外的水平奇偶校验列。

根据另一优选实施方案，Reed Solomon 子空间子码（SSRS）代码将用于对用户数据子块的行编码。该 Reed Solomon 子空间子码代码具体地描述在 M. Hattori, R. J. McEliece, G. Solomon "Subspace subcodes of Reed-Solomon codes", IEEE transactions on IT, vol. 44, no. 5, September 1998。该 SSRS 代码是一种在所有码字中的所有符号的特定比特总为零的代码。与通常的 Reed Solomon 代码相比，需要较少的奇偶校验比特，其代价是附加的用户数据比特。将这种 SSRS 代码应用于哨兵码时，使用伽罗瓦域 GF（2^9）上的 [307，305，3] SSRS 代码。

如果不是独立地对每行编码，而是至少两行一组地对行编码，还可以获得额外的好处，例如首先将三行合并成一个较长的行，然后获得其水平奇偶校验。应用于哨兵码，优选地使用 GF（2^{10}）上的 [916，912，5] RS 代码，需要将每个符号扩展两个比特，优选的是零值比特。

在优选实施方案中，SSRS 代码的使用应用于至少由两个连续行构成的组，这将进一步减少所需的水平奇偶校验的数目。特别是在对哨兵码的应用中，使用了 GF（2^{10}）上的 [917，913，5] SSRS 代码。

附图描述

下面将参考附图详细地对本发明进行解释，其中

图 1 示出哨兵码的编码过程的示意性过程，

图 2 示出根据本发明的编码装置的框图，

图 3 示出根据本发明的编码方法的第一实施方案，

图 4 示出根据本发明的解码装置的框图，

图 5 示出根据本发明的编码方法的第二实施方案，

图 6 示出根据本发明的编码方法的第三实施方案，和

图 7 示出根据本发明的编码方法的第四实施方案。

具体实施例

图 1 示出编码方法的示意性过程，如 WO 00/07300 所描述的。从可以是主机或应用程序的源收到的用户数据首先被分割成数据帧，每帧包括 2048+4 字节；如图 1 中的方框 200 所示，这些帧中的 32 个将在下一编码步骤中使用。在方框 202，组成数据块，并排列成 216 行、304 列的数据块。在方框 204，通过添加 32 个行奇偶校验构成长程码 （LDC）块。在方框 206，ECC 簇排列成 152 列 496 行。这样排列是为了将标号为 ECC 的四个部分填充在物理簇块 218 中，那是全面（comprehensive）代码格式 NTT。

记录系统添加的地址和控制数据也在连续的步骤中转换。首先，逻辑地址和控制数据在方框 208 中排列成 32×18 字节。逻辑地址是那些属于功能性使用的数据，可以指示涉及用户程序的运行时间的方面。同样，物理地址在方框 210 中排列成 16×9 字节。物理地址涉及记录载体上的物理距离，例如光盘。由于重复进行编号和交错，物理和逻辑地址间的联系将被打破。在程序中紧密相随的项将相互间隔开一定的物理距离，反之亦然。同样，映射不是均匀进行的。在方框 212，地址和并到 24 列 30 行的存取方框中。在方框 214，具有 32 个添加了奇偶校验的行。在方框 216，这些被排列成 3 列 496 行的突发指示子码 Burst Indicator Subcode （BIS）簇。这些填充方框 218 中的 3 个 BIS 列。同样，添加了一列同步比特组，这样就构成了 155 列 496 行的物理簇。这些一起构成分组成 496 个记录帧的 16 个物理部分，如图所示。

上述包含 LDC 码字和 BIS 码字的纠错码通常称为哨兵码，并应用于 DVR 技术。关于该代码的进一步细节，特别是该代码的编码和解码设备，代码格式，交错和映射的方法，帧格式，请参考在此引用作为参考的 WO 00/07300。

图 2 示出了根据本发明用于编码，特别是将第三纠错层嵌入图 1 所示的、包含 LDC 和 BIS 两层的装置的框图。图 3 示出了该编码方法的各个步骤对代码的影响。利用这些图，将可以解释将纠错附加层嵌入哨兵码 picket code 的过程。

在第一步骤，来自信息源 10，例如来自传输通道、应用程序或

存储媒体，的用户数据被编码成 304 LDC 码字，该码字是伽罗瓦域 GF（2^8）上的[248，216，33]Reed Solomon 码字。该编码步骤由 LDC 编码器 11 实现，并形成包含用户数据子块 L1 和奇偶校验数据子块 L2 的 LDC 块 L。每个 LDC 码字 c 包含 216 个用户数据符号和 32 个奇偶校验符号，并排列成 LDC 块 L 的列。假定产生随机错误，最有可能的情况是在 304 个 LDC 码字中只有一个 c 导致解码器故障。为了纠正 LDC 码字中的解码器故障，将引入 ECC 第三层。GF（2^8）上的 RS 代码不能直接使用，因为 GF（2^8）上的最长 RS 代码的长度是 255，小于长度为 304 的 LDC 块 L 行的长度。因此，根据本发明，采用 GF（2^9）上的 Read Solomon[306，304，3]代码作为第三层 ELC。

如图 3a 所示，矩阵 V1 用于更进一步的编码，其中 V1 对应于原始码块 L 的用户数据子块 L1。利用比特插入器 12，一个零比特值的比特添加到矩阵 V1 的每个符号，使得每个初始包含 8 个比特的符号现在包含 9 个比特，其中的最后一个比特具有为零的比特值。随后，第三层 ECC 编码器 13 利用 GF（2^9）上的系统的 RS[306，304，3]代码对矩阵 V1 的每行编码。这样，每行产生两个均包含 9 个比特的奇偶校验符号。因而，总共获得了两个附加的奇偶校验符号列 V2。

图 3b 示出如此获得的单个水平码字 h1。码字 h1 包含初始（8 比特）符号 h11，附加的零值比特（h13）和获得的两个均包含 9 个比特的奇偶校验符号 h12。

其后，所产生的水平奇偶校验符号 V2 由第三层 ECC 奇偶校验提取器 14 提取，并进行编码，最后由 BIS 编码器 15 利用[62，30，33]RS 代码写入 BIS 码字。这是可能的，因为在哨兵码的现有格式中，BIS 码字中的 567 个字节没有定义，其中的 486（216×（2×9）/8）个字节用于嵌入所谓的水平奇偶校验。用于编码的其它信息符号由 BIS 信息源 17 传送到 BIS 编码器 15。哨兵码现有格式的其余部分保持不变；第三层只包括所获得的 486 个水平奇偶校验字节。

在交错器 16 中，BIS 码字最终如 WO 00/07300 所述的那样，在交错后的数据流输出到调制器 18 进行进一步处理之前，交错到 LDC 码字中。因为根据本发明提出的第三层与众知的纠错编码和解码方案兼容，DVR 播放器不需要执行第三层的解码。

下面将参考图 4 更加详细地解释对包含上述第三纠错层的扩展

哨兵码的解码，该图示出根据本发明的解码装置的框图。哨兵码通过解码BIS码字、应用擦除策略和解码LDC码字进行解码。下面将进一步解释根据本发明嵌入哨兵码的第三层如何针对 LDC 码字的一次或两次解码故障提供保护的。

首先，利用 BIS-LDC 分离器 21 从由解调器 20 接收到的、包含 BIS 码字和 LDC 码字的数据流中提取 BIS 码字，并利用 BIS 解码器 22 对 BIS 码字解码。其中，只进行唯错误 error-only 解码。因为 BIS 码字受到高纠错能力的保护，所以可以假定所有 BIS 码字都能正确解码。因此，图 3 所示的矩阵 V 的每个 217[219，217，3]RS 码字的两个奇偶校验是已知的。此外，这还提供了关于突发错误的知识，因为突发错误将影响相邻的 BIS 字节，这种识别导致擦除怀疑已发生突发错误的 LDC 字节，即擦除在方框 23 进行。现在，LDC 解码器 24 利用错误-擦除解码策略对 LDC 码字解码，即重构可能包含错误和擦除的代码块 L。

然后，对于用户数据子块 L1，比特插入器 26 插入一个零值比特，将子块 L1 的每个比特扩展一个比特。因此，利用第三层 ECC 解码器 28，所获得的矩阵 V 由 GF（2^9）上的 RS[306，304，3]解码，利用第三层 ECC 奇偶校验提取器 25 提取的水平奇偶校验和从擦除宣告单元 27 获得的关于在初始代码块 L 中的擦除位置的知识，进行解码。根据由 ECC 解码器 28 获得的正确矩阵 V，利用零值比特剥离器 29 将插入的零值比特再次从每个符号中删除，从而获得包含用户数据的用户数据子块 L1，该子块最终输出到信息接收器 30，例如应用程序或传输通道。

ECC第三层可以纠正一些下述的LDC码字的解码器故障和解码器错误：

a）LDC 码字中的两次解码器故障和零次解码器错误：宣告与每个 RS[306，304，3]码字上的两次解码器故障相对应的两次擦除，利用唯擦除（erasure-only）解码对 GF（2^9）上的 RS[306，304，3]码字解码。

b）LDC 码字中的一次解码器故障和零次解码器错误：利用唯错误（error-only）解码对 GF（2^9）上的 RS[306，304，3]码字解码。因为 RS[306，304，3]代码可以纠正一次错误，所以能够恢复解码器

故障。

c）LDC 码字中的零次解码器故障和一次解码器错误：利用唯错误 error-only 解码对 GF（2^9）上的 RS[306，304，3]码字解码。因为 RS[306，304，3]代码可以纠正一次错误，所以能够恢复解码器错误。

本发明解码方法的另一实施方案示于图 5a。其中，利用 GF（2^9）上的 Reed Solomon 子空间子码（SSRS）代码代替了 GF（2^9）上的 RS 代码。SSRS 代码是一种所有码字中的所有符号的某些比特总为零值的代码，即 SSRS 代码是 RS 代码的线性子码 linear subcodes。关于这种 SSRS 代码的更多细节请参考上述的 M. Hattori 等人的文章。

根据图 5a 所使得实施方案，GF（2^9）上的 SSRS[307，305，3]代码是根据从 GF（2^9）上的 RS[306，304，3]代码通过将所有码字中的所有符号的最后一个比特设定为零值而构建的，以每个信息符号一个比特的代价包含奇偶校验符号。这由图 5b 所示的矩阵 V 的一个水平码字 h2 示出。码字 h2 包含初始用户数据字块 L1 的 304 个信息符号 h21，每个符号包含 8 个比特，并向每个符号添加一个比特值为零的比特（h23）。此外，码字 h2 还包含三个附加的符号，每个符号包含作为最后一个比特的、比特值为零的比特，填充最后两个 8-比特符号和倒数第三个符号中的一个附加比特的 17 个奇偶校验比特 h22。倒数第三个符号中的其余 7 个比特 h24 保持为空。

与 GF（2^9）上的 RS[306，304，3]相比，GF（2^9）上的 SSRS[307，305，3]需要更少的奇偶校验比特。GF（29）上的 RS[307，305，3]需要 8×2+1=17 个比特（因为所有包含 h25 的最后一个比特都设定为零值），而 GF（29）上的 RS[306，304，3]需要 9×2=18 个比特。因为矩阵 V 包含 216 个 SSRS[307，305，3]码字，所以利用 GF（2^9）上的 SSRS[307，305，3]代替 GF（29）上的 RS[306，304，3]总共可以节省 1×216/8=27 个字节。然而，如果将 GF（2^9）上的 SSRS[307，305，3]用作 ECC 第三层，那么可以实现相同的纠错能力，因为 GF（2^9）上的 SSRS 代码是 GF（2^9）上的 RS 代码的一个特殊形式。

本发明解码方法的第三实施方案示于图 6a。其中，比 GF（2^9）更大的域用于对一个码字中的几个连续行编码。在第一和第二实施方案中，矩阵 V 的一行构成一个 GF（29）上的 RS[306，304，3]码字，

GF（2^{10}）上的RS[916，912，5]能够将三行编码为一个码字。

为了获得更大的域，需要将两个比特值为零的比特而不是一个添加到用户数据字块L1的每个符号中，从而获得矩阵V1'。通过对每一行编码，可以获得四个附加的奇偶校验符号列V2'，同时形成矩阵V'。

单个水平码字h3示于图6b。每个码字h3包含三个连续的用户数据符号行h311、h312和h313，每行将添加两个零比特值的比特h33，还包含四个10比特的符号，每个通过对三个包含附加的零值h33的连续行h311、h312和h313编码而获得的奇偶校验h32。

与GF（29）上的RS[306，304，3]相比，该代码节省126个奇偶校验字节，并且具有几乎相同的纠错能力。当使用图4所示的类似解码器时，只能纠正LDC码字中的一个解码器故障。一个解码器故障对应于可以纠正四次擦除的GF（2^{10}）上的RS[916，912，5]中的三次擦除。然而，如果LDC码字在第三层的唯错误errors-only解码之后再次解码，那么在许多情况下就可以恢复两次解码器故障或一次解码器错误，因为在假设随机错误的前提下，LDC解码之后剩余的错误符号处于一个GF（2^{10}）上的RS[916，912，5]码字的可能性非常小。此外，当LDC和第三层之间的反复解码次数增加时，将可以获得比单次解码更好的性能。

根据另一可供选择的实施方案，利用GF（2^{10}）上的RS[916，912，5]，可以保护代码块L的所有248行，即用户数据子块L1的所有行和奇偶校验数据子块L2，可以反复对第三层和LDC解码。这将导致比只保护用户数据子块L1更好的性能，其代价是54个额外的奇偶校验字节。在该可供选择的实施方案中，LDC块L的奇偶校验字节也受到第三层的保护，即生成并保护奇偶校验的奇偶校验。奇偶校验的奇偶校验提供了比GF（2^{10}）上的216 RS[916，912，5]还大的完全扩展哨兵码picket code的最小距离，因为第三层和LDC构成一类用在DVD中的乘积码product code。一般而言，具有较大最小距离的乘积码product code可以通过迭代编码实现较好的性能。应当注意的是，该可供选择的实施方案的基本想法，即对即偶校验数据子块L2的行编码，可以应用于任何其它实施方案。

本发明编码方法的另一实施方案示于图7a。该实施方案将示于图5a和图6a的第二和第三实施方案的两种想法结合在一起，即SSRS

代码和对一个码字的多行编码。根据该实施方案，矩阵 V1'的三个连续行同时利用 GF（2^{10}）上的 RS[917，913，5]代码编码，总共获得 38 个奇偶校验字节。

一个这样的码字 h4 示于图 7b。它包括用户数据符号的三个连续行 h411、h412、h413，每行都添加两个比特值为零的比特 h43，所获得的、位于最后五个符号中 38 个奇偶校验比特 h42 和两个空比特 h44。因为也使用 SSRS 代码，所以码字 h4 的最后五个符号的最后两个比特被设定为零值。

在另一可供选择的实施方案中，如果 LDC 块 L 的所有 248 行都受到 GF（2^{10}）上的 SSRS[917，913，5]的保护，如果第三层和 LDC 迭代进行编码，那么将可以获得比 GF（2^{10}）上的 216 SSRS[916，912，5]更好的性能，其代价是 51 个额外的奇偶校验字节。

总之，本发明提高了随机错误的纠错能力，同时保持与现有纠错码方案兼容性。本发明使我们能够选择在冗于性和纠错能力之间的平衡点，并且易于由解码器实现。

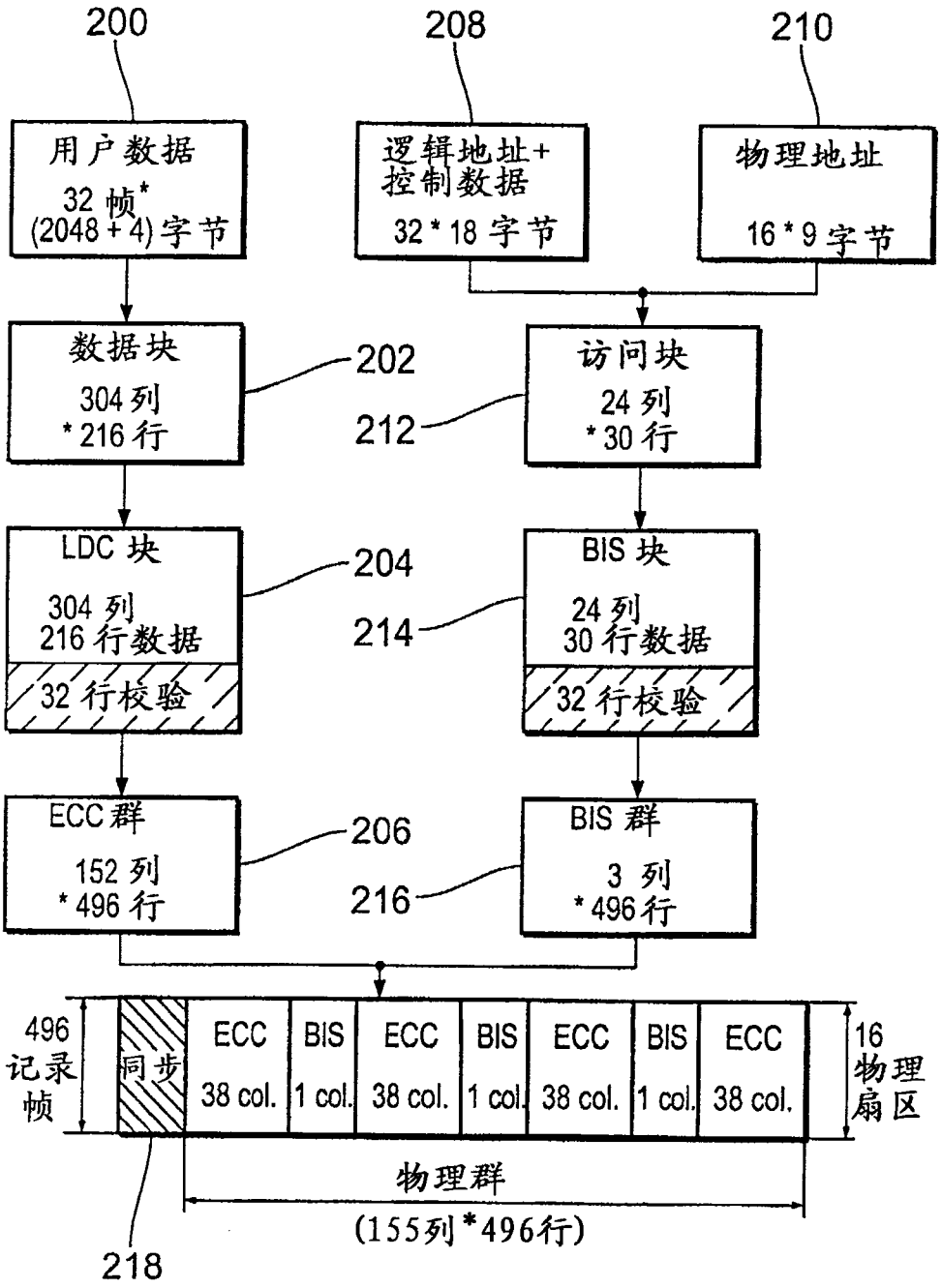

200
208
210
用户数据
32 帧*
(2048 + 4)字节
逻辑地址+
控制数据
32 * 18 字节
物理地址
16 * 9 字节
数据块
304 列
*216 行
202
访问块
24 列
*30 行
212
LDC 块
304 列
216 行数据
32 行校验
204
BIS 块
24 列
30 行数据
32 行校验
214
ECC群
152 列
*496 行
206
BIS 群
3 列
*496行
216
496
记录
帧
同步
ECC
38 col.
BIS
1 col.
ECC
38 col.
BIS
1 col.
ECC
38 col.
BIS
1 col.
ECC
38 col.
16
物理
扇区
物理群
(155列*496行)
218

图 1

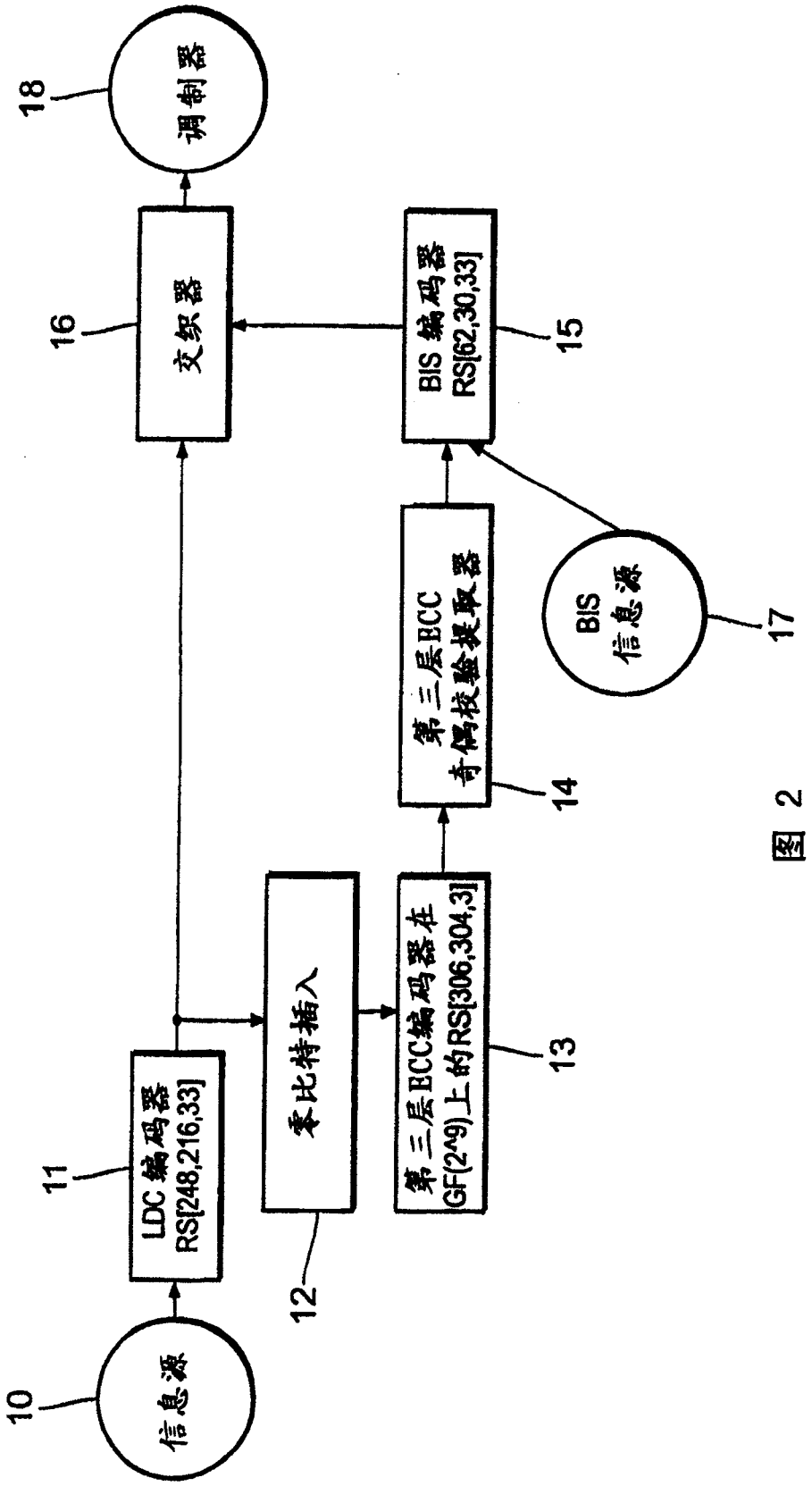

信息源
10
LDC 编码器
RS[248,216,33]
11
零比特插入
12
第三层ECC编码器在
GF(2^9)上的RS[306,304,3]
13
第三层ECC
奇偶校验提取器
14
BIS 编码器
RS[62,30,33]
15
交织器
16
调制器
18
BIS
信息源
17

图 2

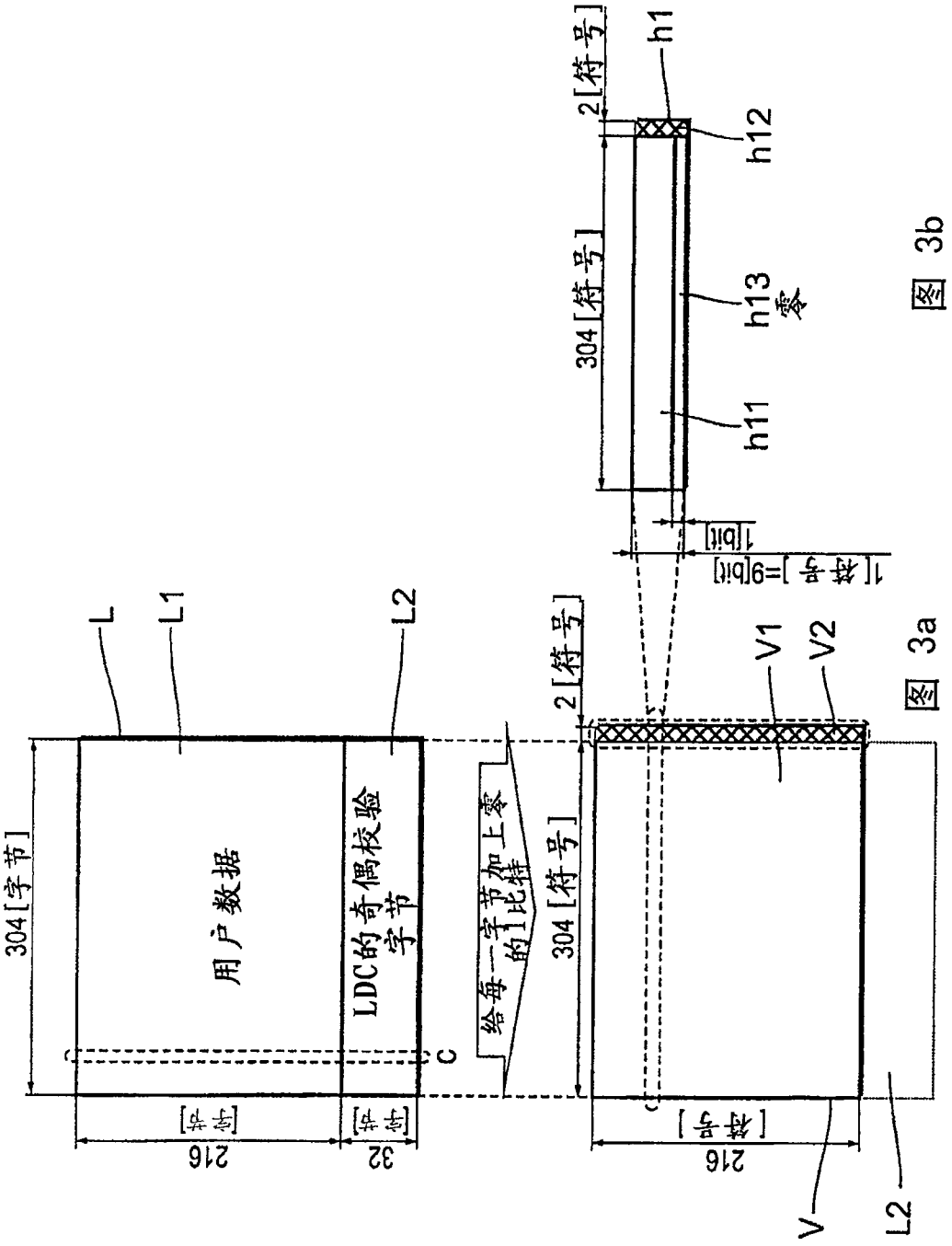

L
L1
L2
304[字节]
用户数据
LDC的奇偶校验字节
c
216[字节]
32[字节]
给每一字节加上零的1比特
2[符号]
304[符号]
216[符号]
V1
V2
V
L2
h1
h12
h13
零
h11
1[比特]
1[符号]=9[比特]

图 3a

图 3b

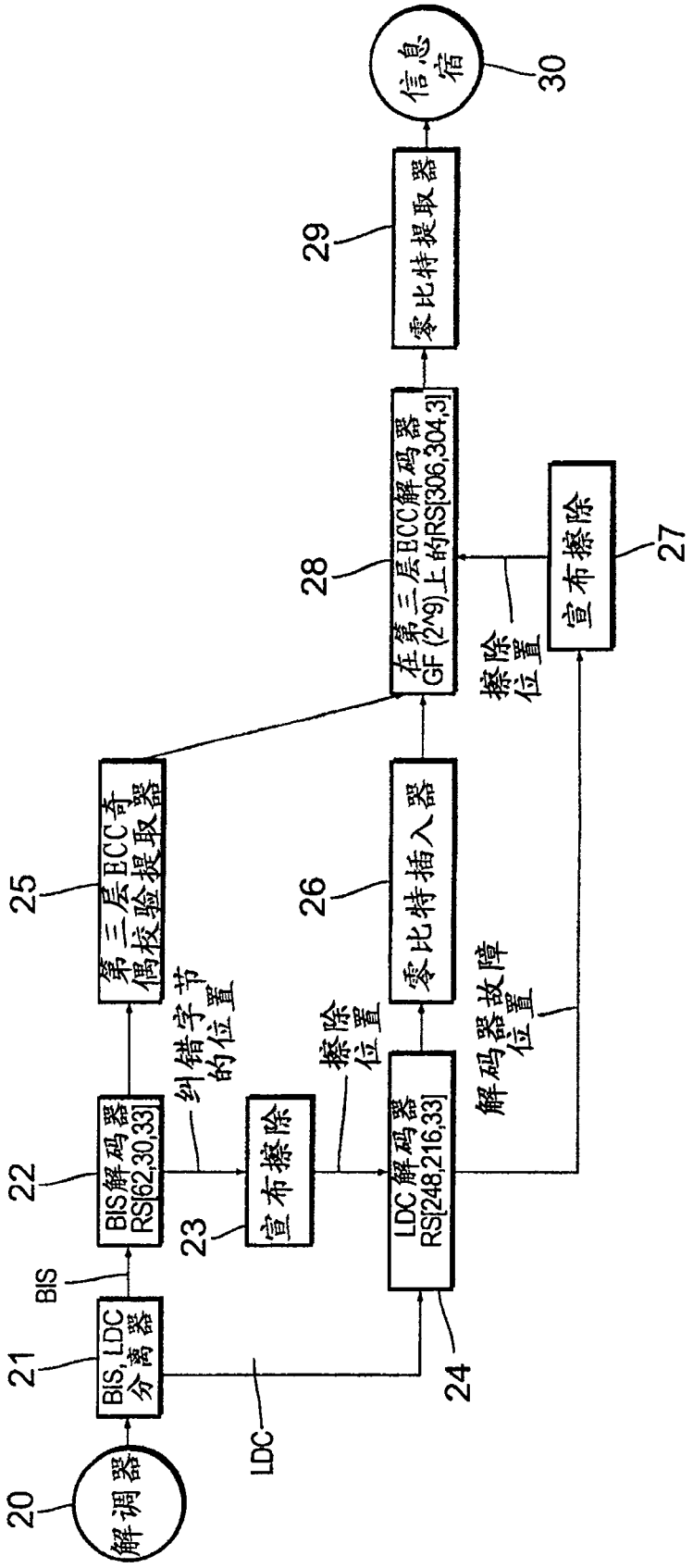

20
解调器
21
BIS, LDC 分离器
BIS
LDC
22
BIS解码器 RS[62,30,33]
纠错字节的位置
23
宣布擦除
擦除位置
24
LDC解码器 RS[248,216,33]
解码器故障位置
25
第三层ECC奇偶校验提取器
26
零比特插入器
27
宣布擦除
擦除位置
28
在第三层ECC解码器 GF (2^9)上的RS[306,304,3]
29
零比特提取器
30
信息宿

图 4

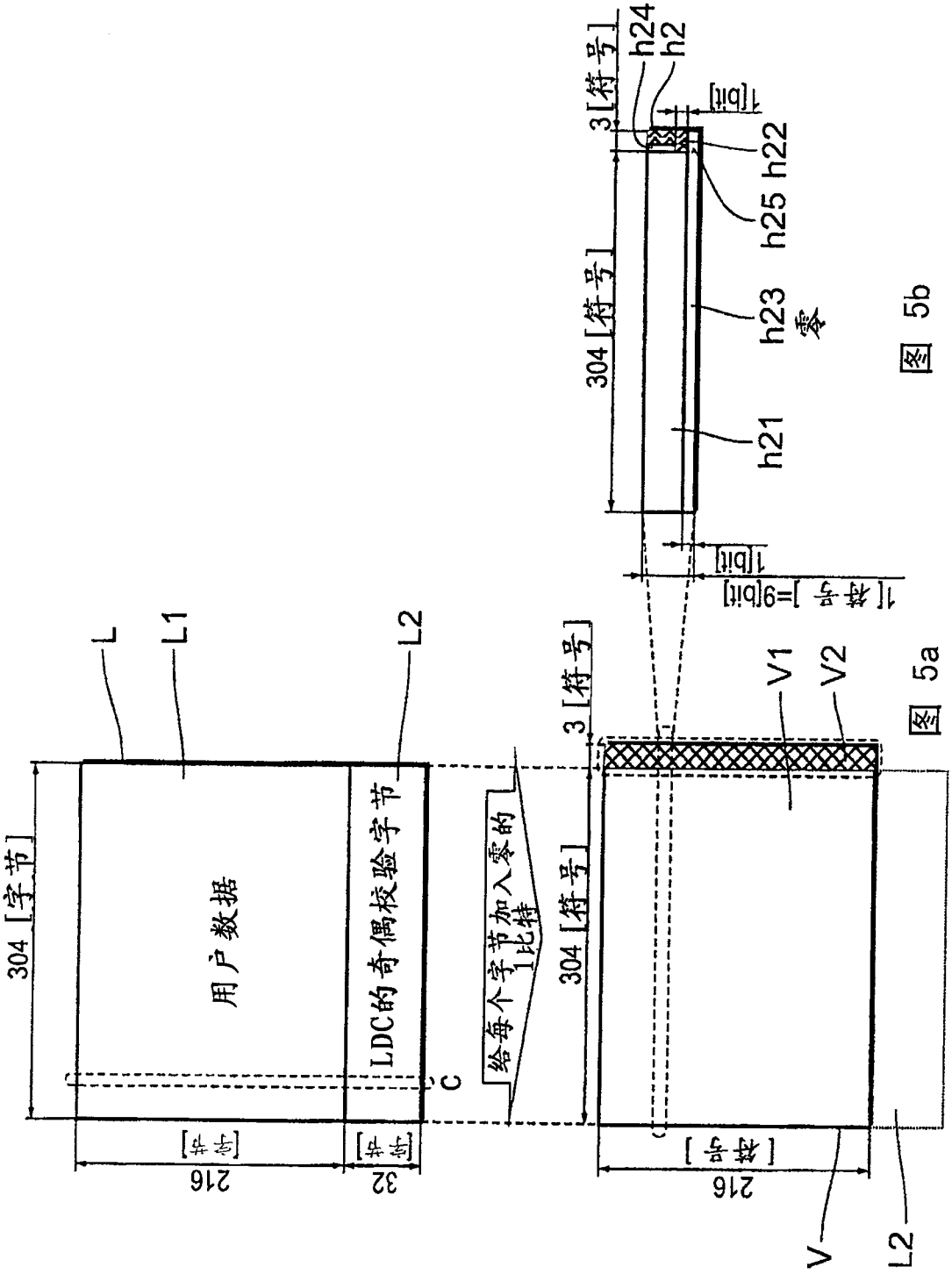

L
L1
L2
304 [字节]
用户数据
LDC的奇偶校验字节
c
给每个字节加入零的1比特
3 [符号]
304 [符号]
V1
V2
V
L2
216 [字节]
32 [字节]
216 [格号]
1[格号]=9[bit]
1[bit]
3[符号]
304 [符号]
h24
h2
h22
h25
h23
零
h21

图 5a

图 5b

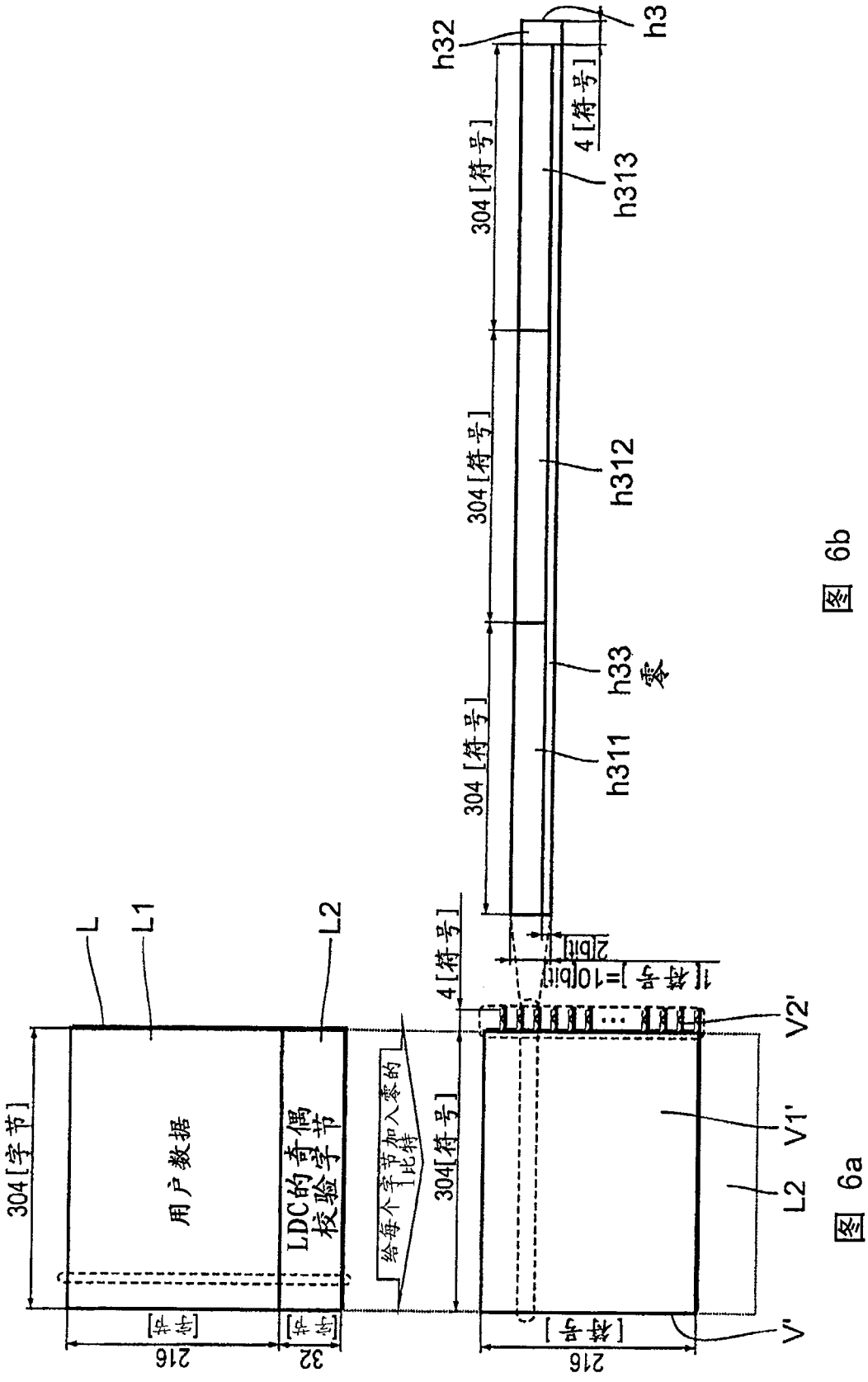

L
L1
L2
304[字节]
用户数据
LDC的奇偶校验字节
216
32
给每个字节加入零的1比特
304[符号]
4[符号]
h32
h3
304[符号]
h313
304[符号]
h312
304[符号]
h33
零
h311
2[比特]
1[符号]=10[比特]
V2'
V1'
L2
V'
216

图 6b

图 6a

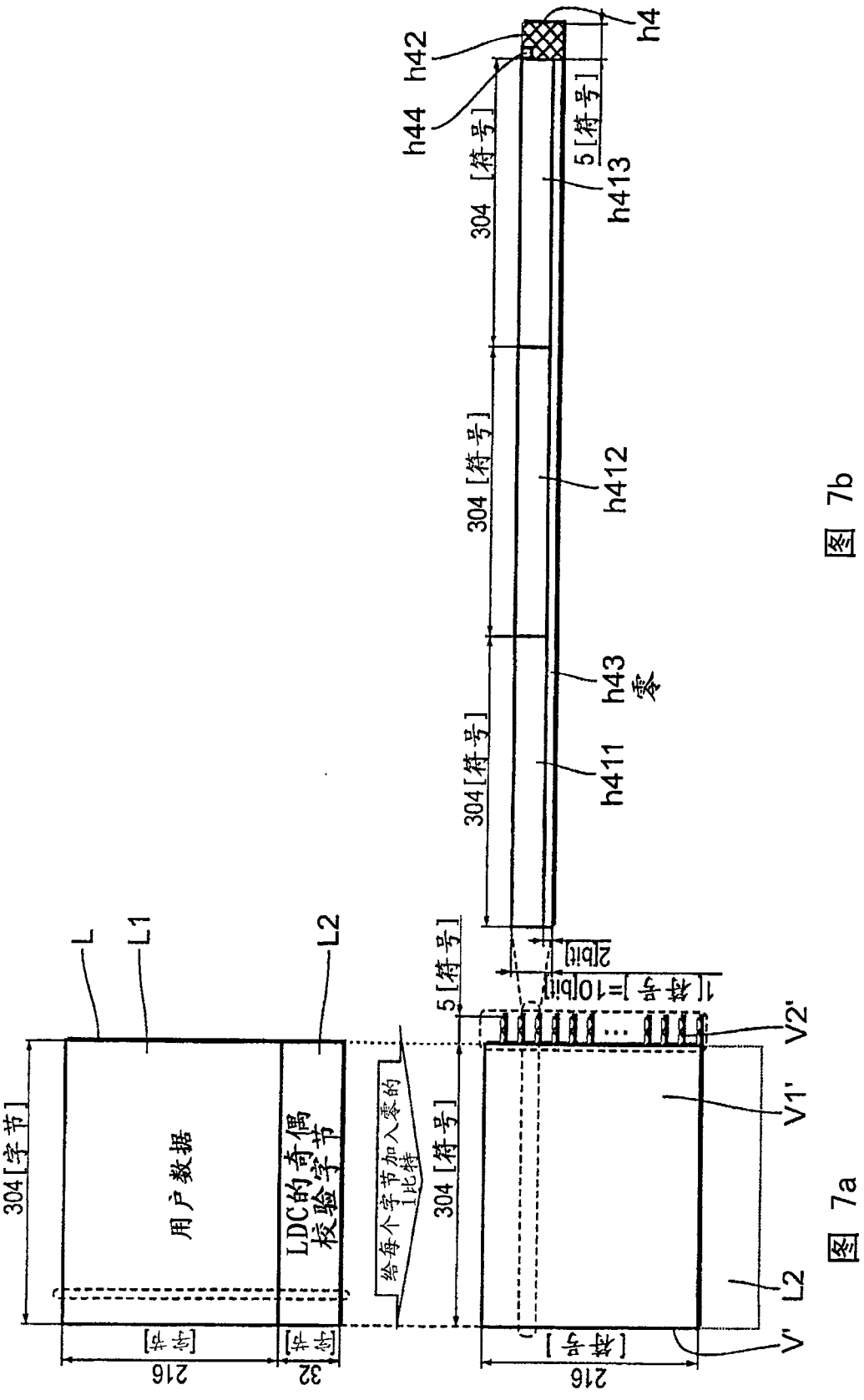

h4
h42
h44
5[符号]
h413
304[符号]
h412
304[符号]
h43
零
h411
304[符号]
5[符号]
20bit
1[符号]=10bit
V2'
V1'
L2
V'
216[符号]
304[符号]
给每个字节加入零的1比特
L
L1
L2
304[字节]
用户数据
LDC的奇/偶校验字节
216[字节]
32[字节]

图 7b

图 7a