



# [12] 发明专利申请公开说明书

[21] 申请号 03814273.2

[43] 公开日 2005 年 8 月 31 日

[11] 公开号 CN 1663128A

[22] 申请日 2003.6.16 [21] 申请号 03814273.2  
 [30] 优先权  
 [32] 2002.6.20 [33] FI [31] 20021221  
 [86] 国际申请 PCT/FI2003/000480 2003.6.16  
 [87] 国际公布 WO2004/001978 英 2003.12.31  
 [85] 进入国家阶段日期 2004.12.20  
 [71] 申请人 诺基亚有限公司  
 地址 芬兰埃斯波  
 [72] 发明人 S·瓦西利亚迪斯 J·尼卡拉  
 J·塔卡拉 P·利乌哈

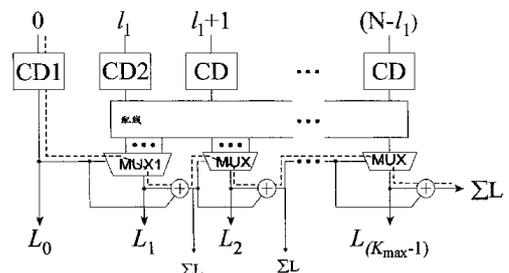
[74] 专利代理机构 中国专利代理(香港)有限公司  
 代理人 程天正 刘杰

权利要求书 3 页 说明书 8 页 附图 5 页

[54] 发明名称 一种用于可变长解码的方法和系统  
以及一种用于代码字定位的设备

### [57] 摘要

本发明涉及一种用于从位流解码可变长代码字的方法，在该位流中为代码字定义最小长度和最大长度。在本方法中，分部分地处理将被解码的位流，在位流的每一部分中搜索代码字，并解码发现的代码字。在本方法中，以这种方式从正处理的位流部分提取至少部分重叠的字段，即：至少两个字段的起点是所述位流部分中代码字的可能的起点。在至少一个字段中，搜索代码字的结尾，并根据代码字终点确定和代码字相关的数据。使用和至少一个代码字相关的数据来发现字段中将被解码的代码字的发生，并解码发现的代码字。本发明还涉及解码器、其中应用本方法的系统和设备，以及存储应用本方法的程序的存储装置。



1. 一种从位流解码可变长代码字的方法，在该位流中为代码字定义最小长度和最大长度，在本方法中分部分地处理将被解码的位流，对位流的每一部分进行代码字的搜索，并解码发现的代码字，其特征在于，在本方法中，以这种方式从将被处理的位流部分提取至少部分重叠的字段，使得至少两个字段的起点是所述位流部分中代码字的可能的起点，从至少一个字段搜索代码字结尾，根据代码字终点确定和代码字相关的数据，并使用和至少一个代码字相关的数据来确定字段中将被解码的代码字的发生，以及解码发现的代码字。
2. 如权利要求 1 所述的方法，特征在于和将被解码的代码字相关的数据包括将被解码的代码字的长度，其中搜索将被解码的代码字以如下方式执行：
- 在第一字段中搜索将被解码的第一代码字，
  - 在正处理的位流部分中，在其起点是对应将被解码的在前代码字长度之和的点的字段中，搜索将被解码的下一代码字，
  - 执行上述步骤以在正处理的位流部分中发现所有将被解码的代码字。
3. 如权利要求 2 所述的方法，特征在于在本方法中，为了发现将被解码的下一代码字的起点，计算将被解码的在前发现的代码字长度之和并基本上同时选择该起点。
4. 如权利要求 3 所述的方法，特征在于在从正处理的位流部分发现代码字中，使用下面的公式：
- $$O = P_0 \bar{s}_2 \bar{s}_1 \bar{s}_0 + P_1 \bar{s}_2 \bar{s}_1 s_0 + P_2 \bar{s}_2 s_1 \bar{s}_0 + P_3 \bar{s}_2 s_1 s_0 + P_4 s_2 \bar{s}_1 \bar{s}_0 + P_5 s_2 \bar{s}_1 s_0 + P_6 s_2 s_1 \bar{s}_0 + P_7 s_2 s_1 s_0$$
- 其中， $s_2, s_1, s_0$  是在正检查的字段之前发现的、将被解码的代码字长度之和的位，
- $P_0-P_7$  表示在正处理的位流部分中从位置  $p_i$  开始的那些代码字的所有可能的代码字长度，以及
- $O$  是在不同的备选对象中对将被搜索的代码字的结尾的选择。
5. 如权利要求 1 至 4 任一个所述的方法，特征在于至少第一个、第二个和第三个发现的代码字被解码。
6. 一种从位流解码可变长代码字的解码器，在该位流中为代码字定义最

小长度和最大长度，所述解码器配有：用于以各定长的部分处理将被解码的位流的装置，用于在位流的每一部分中搜索代码字的装置，和用于解码发现的代码字的装置，特征在于解码器还包括：用于以这种方式从正处理的位流部分提取至少部分重叠的字段的装置，使得至少两个字段的起点是所述位流部分中代  
5 码字的可能的起点；用于在至少一个字段中搜索代码字结尾的装置；和用于根据该结尾来确定和在该字段中发现的代码字相关的数据的装置（CD）；以及用于根据和至少一个代码字相关的数据，确定该字段中将被解码的代码字的发生的装置（MUX1，MUX，M，MA）。

7. 如权利要求 6 所述的解码器，特征在于和将被解码的代码字相关的数据包括将被解码的代码字的长度，以及该解码器包括：

- 用于在第一字段搜索将被解码的第一代码字的装置（CD1），
- 用于搜索该字段中将被解码的下一代码字的搜索装置（CD，MUX1，MUX，M，MA），其中安排所述装置以形成在字段中发现的代码字的长度数据，
- 15 • 用于在起点是将被解码的在前代码字长度之和的位置处从正处理的位流部分选择将被解码的代码字的选择装置（MUX1，MUX，MA）。

8. 一种传输二进制形式的信息的系统，包括一个编码器和一个解码器，所述解码器包括用于从位流解码可变长代码字的装置，在该位流中为代码字定义最小长度和最大长度，所述解码器配有：用于以各定长的部分处理将被解码  
20 的位流的装置，用于在位流的每一部分搜索代码字的装置，和用于解码发现的代码字的装置，特征在于解码器还包括：用于以这种方式从正处理的位流部分提取至少部分重叠的字段的装置，使得至少两个字段的起点是所述位流部分中代码字的可能的起点；用于在至少一个字段中搜索代码字结尾的装置；和用于根据终点确定和在该字段中发现的代码字相关的数据的装置；以及用于根据和  
25 至少一个代码字相关的数据来确定该字段中将被解码的代码字的发生的装置。

9. 如权利要求 8 所述的系统，特征在于和将被解码的代码字相关的数据包括将被解码的代码字的长度，以及所述解码器包括：

- 用于在第一字段搜索将被解码的第一代码字的装置（CD1），
- 用于在该字段中搜索将被解码的下一个代码字的搜索装置（CD，MUX1，MUX，M，MA），其中安排所述装置以形成在该字段中发现的代码字的长度数据，
- 30 • 用于在起点是将被解码的在前代码字的长度之和的位置处从正处理的位

流部分选择将被解码的代码字的选择装置 (MUX1, MUX, MA)。

10. 一种包括从位流解码可变长代码字的解码器的设备, 在该位流中为代码字定义最小长度和最大长度, 所述解码器配有: 用于以各定长的部分处理将被解码的位流的装置, 用于在位流的每一部分中搜索代码字的装置, 和用于解  
5 码所发现的代码字的装置, 特征在于, 解码器还包括: 用于以这种方式从正处理的位流部分提取至少部分重叠的字段的装置, 使得至少两个字段的起点是所述位流部分中代码字的可能的起点; 用于在至少一个字段中搜索代码字结尾的装置; 以及用于根据终点确定和在该字段中发现的代码字相关的数据的装置; 以及用于根据和至少一个代码字相关的数据来确定该字段中将被解码的代码字  
10 发生的装置。

11. 如权利要求 10 所述的设备, 特征在于和将被解码的代码字相关的数据包括将被解码的代码字的长度, 以及该解码器包括:

- 用于在第一字段中搜索将被解码的第一代码字的装置 (CD1),
- 用于在该字段中搜索将被解码的下一代码字的搜索装置 (CD, MUX1,  
15 MUX, M, MA), 其中安排所述装置以形成在该字段中发现的代码字的长度数据,

- 用于在起点是将被解码的在前代码字的长度之和的位置处从正处理的位流部分选择将被解码的代码字的选择装置 (MUX1, MUX, MA)。

12. 如权利要求 10 或 11 所述的设备, 特征在于该设备包括对将被解码的代码字长度求和的装置 (MA), 以发现将被解码的代码字的起点并基本上同时选择  
20 该起点。

13. 如权利要求 10、11 或 12 所述的设备, 特征在于该设备包括用于执行移动通信的装置。

14. 一种存储程序的存储介质, 包括计算机可读指令, 执行该指令使计算  
25 机从位流解码可变长代码字, 在该位流中为代码字定义最小长度和最大长度, 该程序配有: 以定长的部分处理将被解码的位流的计算机指令, 在位流的每一部分中搜索代码字的计算机指令, 和解码所发现的代码字的计算机指令, 特征在于, 该程序包括: 以这种方式从正处理的位流部分提取至少部分重叠的字段的计算机指令, 使得至少两个字段的起点是所述位流部分中代码字的可能的起  
30 点; 在至少一个字段中搜索该代码字结尾的计算机指令; 和根据终点确定与在该字段中发现的代码字相关的数据的计算机指令; 以及根据和至少一个代码字相关的数据来确定在该字段中将被解码的代码字发生的计算机指令。

一种用于可变长解码的方法和  
系统以及一种用于代码字定位的设备

5

本发明涉及一种从位流中解码可变长代码字的方法，在该位流中为代码字定义最小长度和最大长度，在本方法中，分部分地处理将被解码的位流，在位流的每一部分中进行代码字搜索，并且解码发现的代码字。本发明还涉及一种用于从位流中解码可变长代码字的解码器，在该位流中为代码字定义最小长度和最大长度，所述解码器配有用于以特定长度的部分处理将被解码位流的装置，用于在位流的每一部分搜索代码字的装置，以及用于解码发现的代码字的装置。本发明还涉及一种传输二进制形式的信息的系统，该系统包括一个编码器和一个解码器，并且所述解码器包括用于从位流解码可变长代码字的装置，在该位流中为代码字定义最小长度和最大长度，所述解码器配有用于以定长部分处理将被解码该位流的装置，用于在位流的每一部分搜索代码字的装置，以及用于解码发现的代码字的装置。而且，本发明涉及一种包括从位流解码可变长代码字的解码器的设备，在该位流中为代码字定义最小长度和最大长度，所述解码器配有用于以定长部分处理将被解码的位流的装置，用于在位流的每一部分中搜索代码字的装置，以及用于解码发现的代码字的装置。最后，本发明涉及用于存储程序的存储设备，所述程序包括可由机器执行的、从位流中解码可变长代码字的步骤，在该位流中为代码字定义最小长度和最大长度，所述程序配有以特定长度部分处理将被解码的位流的计算机指令，在位流的每一部分搜索代码字的计算机指令，以及解码发现的代码字的计算机指令。

在通过数据网络传输图像和视频数据中，需要压缩图像数据，这是因为由可获得的传输速率来看，将被传输的数据量是非常大的。已知的压缩算法包括无损算法和有损算法。离散余弦变换（DCT）是结合量化，用于从图像数据去掉冗余信息的有损算法。为基本图像单元（例如， $8 \times 8$  象素）计算 DCT 因子，并且图像块以由这些因子组成的矩阵表示。在 DCT 后，通过锯齿形（Zig-Zag）扫描将这些因子排列成合适的顺序用于无损哈夫曼编码。哈夫曼编码仅是可变长编码（VLC）的一个示例，可变长编码（VLC）在 DCT 和量化之后降低统计冗余。可变长编码的原理是用较短的代码字表示频繁发生的符号而用较长的

代码字表示不常发生的符号。为了开始编码，要通过根据符号发生的概率安排各个符号来处理源数据。概率最小的符号被递归地组合到一个符号中，其中形成一个树结构（如图 1a 所示的示例）以通过定义一位来代表每个分支而为每个符号分配它自身的代码字。从图 1a 的代码树可以看出，发生概率最小的符号具有最长的代码字。表 1b 表示每个符号的发生概率  $P(X)$ ，代码字（代码）和熵  $H(x)$ 。上述方法导致对于平均值的最短的代码字，并且上述压缩非常接近于原始源的熵；因此，编码在总体上是有效的。从而，可变长编码在许多涉及图像处理的标准中是一个必要的元素。

如可变长编码的定义指出的，可变长编码的代码字在长度上是变化的。如果不能从位流直接检测到代码字的起点和终点，在代码字间会产生递归依赖性，这会使解码更困难。这种依赖性将使得在可变长解码（VLD）前必须从位流发现代码字，也就是说必须确定代码字的位置。因此，在解码过程中，确定码长和位置、代码字和相应的符号是必须的。

根据下面方式的位流处理可表征 VLD 实现。在位的串行处理中，连续地一位一位的处理位流。因此，整个进入的位流是以恒定的输入速率处理的。因为代码字具有可变长度，代码字的全部解码也用去可变的时间，这将导致可变的输出速率。

在位的并行处理中，以恒定的输入速率或可变输入速率并行处理进入位流的多个位。当输入速率保持恒定时，输出速率仍然是可变的，这是因为代码字的可变长度而导致的。为获得恒定的输出速率，需要可变地处理进入位流。换句话说，为保证恒定的输出速率，输入必须包含对应于所需输出速率和最长代码字长度乘积的多个位。

以类似于根据位流处理的特征的方式，解码器的并行性也可以用将被同时解码的多个代码字（或从输出获得的符号）来描述。当一次对一个代码字执行解码时，解码可以被表征成符号串。因此，当解码结果是一次获得多个符号时，解码是符号并行的。

解码可以通过基于树的算法实现，该算法是与哈夫曼树结构相反的实现：将编码的进入位流与二叉树进行比较，从根开始，并一直执行直到从各个叶单元确定了整个代码字。在树模型的解码中，只有用短代码字才能获得有效解码。然而，对实时处理的高要求需要对长代码字也应该获得相应的输出速率。而且，因为树模型的位串行形式，所以树模型不适合于同时解码几个符号，即，不适合用于符号的并行处理，这是因为当处理单个位流时代码字之间的依赖性。

为了改进处理能力，已通过并行处理多位来提高并行性，这是通过在输入缓冲器中收集足够的数据量（最大代码字长度）而实现的，从缓冲器的开始检测和解码代码字。在符号串行解码中，为确保每一循环发现一个符号，在输入缓冲器中累积对应最大代码长度的数据量。这里的问题在于多个位被留在缓冲器的末端，它们不能在当前循环中被利用。符号并行解码（或多符号解码）涉及设计方面的问题和挑战：装置或系统将很快变得非常复杂。

Hsieh & al 在 C. T. Hsieh 和 S. P. Kim “A concurrent memory-efficient VLC decoder for MPEG applications” IEEE Trans Consumer Electron. , 42 卷第 3 期 439-446 页, 1996 年 8 月中给出在同一循环中同时并行解码短代码字（并行解码算法）。解码器使用可变长编码的概率分布，根据该概率分布，进入位流的位序列有较大概率包括比长代码字更短的代码字。例如，在最差的情况下，一个 16 位进入数据序列可包含 8 个 2 位代码字，这些代码字的一次一个符号的解码需要 8 次解码循环。在出版物中公开的解码算法的目标是通过同时检测 2 个或更多较短代码字而加速解码过程；换句话说，使用该方法来检测代码字的组合。例如，在解码两级代码字中，树模型的第一级对应将被解码的第一代码字，以及第二级对应将被解码的第二代码字。因此，目标是同时解码两个代码字。第一代码字的长度可能在 2 至  $k$  位之间变化。如果第一代码字的长度是 2 位，那么可能被并行解码的第二代码字的长度是 2 至  $k_2$  位。另一方面，如果第一代码字的长度是 3 位，那么可能被同时解码的第二代码字的长度是 2 至  $k_3$  位等等。因为第一级包括所有代码，因此  $k$  等于代码本的最大长度，对应最长代码字的长度。在第二级，根据 VLC 的基本属性优选较短的（较可能）代码。因此， $k_i$  的值在范围  $0 \leq k_i \leq k$  内。

两级解码过程中的基本过程开始于树模型的根。对照每级上所有可能的代码字，大致上同时测试树中每个分支的位序列。如果在树的第二级上发现一个以上的合适分支，那么将选择从根一直连续的路径，也就是，位流是连续而无中断的。如果下一级没有分支给出从根的连续路径，则解码过程可在第一级结束。这意味着不能解码第二级代码字。使用从每级获得的代码字长度来控制输入流的下一个数据，其中解码过程将再次开始。也可以对更多级使用相同的解码方法。根据最大可能位模式（MLBP）的代码字检测块返回 3 个因子以完成解码循环：确定组中残余的组标记，转移位流到下一个代码字开始的长度，以及用于存储器的组代码。根据残余和组来从随机存取存储器检索该符号。

本发明的目的是通过在解码过程的第一步中去除代码字间的依赖性，而使

得可能并行地且管道化地进行解码，在解码过程的第一步中定位该代码字。在根据本发明的方法中，从输入缓冲器中提取定长的位字段，以发现所有可能的代码字。借助级联结构从缓冲器大致同时检测所有可能的代码字。换句话说，从进入缓冲器的位流中，从每个可能的缓冲器位置检测代码字长度。然而，这

5 将导致从输入缓冲器中也检测到无效的代码字。在根据本发明的方法中，通过使用第一检测器（第一代码字长度）控制第二输出的选择的方式，根据在前的代码字长度而选择有效的代码字长度。依次，根据在前的代码字长度之和选择第三代码字长度等等。在选择实际代码字中，使用多路复用器从几个代码字长度（这些长度已从缓冲器中所有可能的代码字位置获得）选择有效的代码字长度。

10 例如，可以使用基本逻辑电路（与/或门和反相器）实现上述多路复用器。在根据本发明的方法中，多路复用器以这样的方式链接，它们给出缓冲器中所有完整的代码字的长度作为它们的直接输出。关于部分和以及所检测代码字长度之和的数据也可获得以稍后使用。本发明的一个优选实施例的特征在于实施级联的结构，其通过 MultiplexedAdd(MA 复用相加)设备使关键路径更短而变得

15 更长。在本发明的方法中，可以检测缓冲器中出现的所有完整代码字；因此，可以并行执行对符号的实际搜索，以在没有限制依赖性的情况下加速解码。

为使之更为精确，根据本发明的方法的主要特征在于，在本方法中，从将被处理的部分位流中以这样的方式分离出至少部分重叠的字段，使得至少两个字段的起点是所述位流部分中代码字的可能的起点，从至少一个字段搜索代码

20 字结尾，以及根据该代码字的终点确定与代码字相关的数据，并且使用和至少一个代码字相关的数据来确定该字段中将被解码的代码字的发生，并解码发现的代码字。根据本发明的解码器的主要特征在于，所述解码器也包括：从正处理的位流部分提取至少部分重叠的字段的装置，使得至少两个字段的起点是所述位流部分中代码字的可能的起点；在至少一个字段中搜索代码字结尾的装置；和根据该结尾确定与该字段中发现的代码字相关的数据的装置（CD）；以及根据和至少一个代码字相关的数据，确定字段中将被解码的代码字的发生的装置（MUX1, MUX, M, MA）。根据本发明的系统的主要特征在于，解码器也包括：用于以这种方式从正处理的位流部分提取至少部分重叠的字段的装置，使得至少两个字段的起点是所述位流部分中代码字的可能的起点；用于在

25 至少一个字段中搜索代码字结尾的装置；和用于根据终点确定和在该字段中发现的代码字相关的数据的装置；以及用于根据和至少一个代码字相关的数据来确定该字段中将被解码的代码字的发生的装置。根据本发明的设备的主要特征

在于，解码器也包括：用于以这种方式从正处理的位流部分提取至少部分重叠的字段的装置，使得至少两个字段的起点是所述位流部分中代码字的可能的起点；用于在至少一个字段中搜索代码字结尾的装置；和用于根据终点确定和在该字段中发现的代码字相关的数据的装置；以及用于根据和至少一个代码字相关的

5 数据来确定该字段中将被解码的代码字的发生的装置。最后，根据本发明的存储装置的主要特征在于，程序包括：以这种方式从正处理的位流部分提取部分重叠的字段的计算机指令，使得至少两个字段的起点是所述位流部分中代码字的可能的起点；在至少一个字段中搜索代码字结尾的计算机指令；和根据终点确定和在该字段中发现的代码字相关的数据的计算机指令；以及根据和至少

10 一个代码字相关的数据来确定该字段中将被解码的代码字的发生的计算机指令。

本发明通过位并行、符号并行的方式改进了解码代码字的现有技术，该方式也适合于并行长代码字的解码。因此，本发明可以使用较长的输入缓冲器（任何长度的缓冲器）。而且，本方法将解码划分成阶段以便使该实施流水线化。

15 在代码字可能位于的输入缓冲器的所有位置搜索可能的代码字。有效代码字通过索引来检测，上述索引是通过计算在前适合代码字的长度之和获得的。因为关于限制（起点和终点）的数据是与发现代码字相关地发现的，所以独立于其他代码字来解码该代码字并提高并行性是可能的。根据本发明的方法的输出阶段与输入缓冲器的大小成比例。

20 下面，将参考附图更详细的描述本发明，其中  
 图 1a 表示通过哈夫曼编码形成的树结构，  
 图 1b 以表格的形式表示哈夫曼代码字和相应的符号以及发生的概率，  
 图 2 表示根据优选实施例从位流发现 VLC 代码字的原理，  
 图 3 以简化的图表表示根据发明的优选实施例从位流发现 VLC 代码字的

25 系统，

图 4a 表示图表中的 MultiplexedAdd 装置，  
 图 4b 表示 MultiplexedAdd 的内部结构，以及  
 图 5 以简化的图表表示根据发明的另一优选实施例通过使用 MultiplexedAdd 从位流发现 VLC 代码字的系统。

30 让我们首先定义在缓冲器中一次可出现多少可变长代码字。为此，代码表的代码长度由组  $S_L = \{l_1, \dots, l_n\}$  定义，其中  $l_1$  和  $l_n$  分别表示代码字的最小长度和最大长度。因此，在一个  $N$  位缓冲器中，代码字的最大长度是  $K_{\max} = \lfloor N/l_1 \rfloor$ ， $N \geq l_n$ 。

让我们指出具有符号  $W_i$  的可变长代码字, 其中  $i=0,1,\dots,(K_{\max}-1)$ , 和具有符号  $L_i$  的代码字  $W_i$  的长度。同时, 让我们定义索引  $j_i$ ,  $0 \leq j_i \leq (N-1)$ , 来表示在  $N$  位的代码字缓冲器中代码字  $W_i$  的第一位。

一般而言, 可以假设第一个代码字  $W_0$  总是位于缓冲器的开始, 其中  $W_0$  的起点是  $j_0=0$ 。第二个代码字  $W_1$  正位于第一个代码字之后, 其中定义第二个代码字的起点的索引  $j_1$  是第一个代码字的长度, 也就是,  $j_1=L_0$ 。这意味着代码字  $W_i$  的起点是在前代码字的长度之和, 也就是,  $j_i = \sum_{k=0}^{i-1} L_k$ 。

缓冲器中代码字长度并非是事先已知的。为避免递归的依赖性, 也需要在缓冲器中任意位置并行搜索代码字。一般而言, 任何代码字的所有可能位置可以由  $p=\{1,1+1,1+2,\dots,N-1\}$  定义, 这在  $N$  位缓冲器中意味着  $N-2(l_1-1)$  个位置。因为代码字的最大长度  $l_n$  是已知的, 将不大于最大长度  $l_n$  大小的字段从  $p$  定义的所有可能位置分离, 通过使用例如模式匹配、代码表属性, 诸如引导字符、最大可能位模式 (MLBP) 或数字属性而在上述的每个字段中搜索可能的代码字长度。然而, 只有定义代码字的所有位可获得来完成解码, 即发现符号, 才能发现符号。为此原因, 在以最后  $l_n-1$  索引开始的字段中检测代码字是比较容易的; 仅需要从以起点  $N-K$  开始的位字段搜索  $K$  位代码字, 其中  $K < l_n$ 。

在上述过程中, 检测冗余代码字, 这是因为如果从有效较长代码字的中间分离出位字段, 则可能在此代码字中发现更短的代码字。为此原因, 每次搜索过程只返回所检测代码字的长度。缓冲器中有效代码字的索引可以递归性地由长度定义; 第一代码字长度定义第二代码字的起点; 第一和第二代码字的长度定义第三代码字的起点, 等等。

图 2 表示通过根据本发明的一个优选实施例的方法, 在 16 位缓冲器中发现代码字的示例。让我们假设其中长度由组  $S=\{2,3,4,5,6,7,8\}$  定义以及 16 位输入缓冲器中代码字的最大长度是  $K_{\max}=8$  的代码表。根据上述方法, 从缓冲器中区分 14 个字段  $F_0-F_{13}$ , 在每个字段中尝试发现可能的代码字。确定的是, 第一字段  $F_0$  包括第一合适的代码字  $W_0$ 。在字段  $F_1-F_7$  中的一个发现第二代码字  $W_1$ 。以相应的方式, 可以在字段  $F_3-F_{13}$  之一中发现可能的第三代码字  $W_2$ 。可以在字段  $F_5-F_{13}$  中发现可能的其他代码字  $W_3-W_7$ 。在字段  $F_8-F_{13}$  中的位字段长度比在其他字段中的长度短, 这是因为装在缓冲器末端的代码字比最大长度  $l_n$  要短。

为了完成解码过程, 从代码表检索对应代码字的符号。因为可以通过上述

方法发现代码字的起点和终点，所以消除了代码字之间的依赖性。换句话说，可以从输入流分离出代码字，并可独立地执行搜索。

总体上，描述的模式可以如下表示：

- 确定  $N$  位代码字缓冲器中代码字的最大数量  $K_{\max}$ ，
- 5     • 从缓冲器分离必要大小的  $(N-2(l_1-1))$  个字段  $F_0-F_{(N-2(l_1-1)-1)}$  以确定代码字长度。各字段的起点是缓冲器位置  $\{0, l_1, l_1+1, l_1+2, \dots, N-l_1\}$ ，
- 在每个字段的开始处发现可能的代码字。返回所发现代码字的长度，
- 在第一字段发现第一代码字的长度，通过索引在缓冲器发现其他有效的代码字，该索引通过计算在前合适代码字的长度之和而获得，
- 10    • 根据代码表发现对应于合适的代码字的符号。

因为在上述方法中输入级别和输出级别是可变的，所以在输入和输出中将需要缓冲。图 3 表示实现上述过程的解码器，在解码器的设计中已经考虑到同时检测输入缓冲器中的所有代码字。为了实现该目的，使用  $N-2(l_1-1)$  个代码字检测器 CD。第一代码字检测器 CD1 (图 3 中左边一个) 使用从输入缓冲器  
15 开端开始的字段以检测代码字。第二代码字检测器 CD2 使用从输入缓冲器的点  $l_1$  开始的字段以检测代码字。在根据该图的顺序中，左面的  $N-l_1-l_1+2$  个代码字检测器使用从输入缓冲器不同点开始的字段 (最大长度代码字)，而其他  $l_1-l_1$  个代码字检测器只需要检测比最大长度短的代码字。所有的代码字检测器 CD1—CD  $(N-2(l_1-1))$  大致上同时检测代码字并返回所发现代码字的长度。为了  
20 从所有发现的代码字选择合适的代码字长度  $L_i$ ，使用多路复用器 MUX，其形成一个级联结构。每个多路复用器 MUX 具有来自每个代码字检测器 CD 的输入，检测器的位置由组  $p$  定义。代码字检测器 CD 的位字段在输入缓冲器的位置  $il_1-il_n$  中。因为第一代码长度  $L_0$  总是从第一代码字检测器 CD1 获得，所以它将控制第一多路复用器 MUX1 去选择第二合适代码长度  $L_1$ 。这也可以被用于  
25 转换关于解码模式的信息。换句话说，如果代码长度是 0，则或者终止解码或者已经发现错误。由在前代码长度之和控制其他多路复用器。以这种方式，代码长度之和的计算产生一条关键路径，在图 3 中该路径由虚线表示。

根据长度数据可以将代码字从输入缓冲器中分离出来并独立解码。而且，当已去除代码字的依赖性时，可以并行进行解码。

30 在根据本发明另一优选实施例的系统中，使用 MultiplexedAdd MA，其示意图如图 4a 所示，且内部结构图如图 4b 所示。可以使用该方案来减小在确定代码字中产生的关键路径。如图 4a 所示，MultiplexedAdd MA 计算两个输入之

和并且也大致上同时执行复用，图 4b 表示 3 位 MultiplexedAdd 的内部结构。一个 X 位 MultiplexedAdd 包括 X 个全加器 FA 和一个 X 级别复用树结构，该结构接收全加器之和作为其输入，作为选择信号以及备选对象（概率，P）。如图所示的系统示例是两个 3 位数字 A 和 B，它们的和 S 控制从概率 P0-P7 的输出 O 的选择（对应从概率 P 选择的代码字长度  $L_i$ ），在根据本发明的方法中，  
5 这些概率代表从输入缓冲器的位置  $p_i$  开始的那些代码字的所有可能的代码字长度。因此，实现输出的公式可以定义如下：

$$O = P_0 \bar{s}_2 \bar{s}_1 \bar{s}_0 + P_1 \bar{s}_2 \bar{s}_1 s_0 + P_2 \bar{s}_2 s_1 \bar{s}_0 + P_3 \bar{s}_2 s_1 s_0 + P_4 s_2 \bar{s}_1 \bar{s}_0 + P_5 s_2 \bar{s}_1 s_0 + P_6 s_2 s_1 \bar{s}_0 + P_7 s_2 s_1 s_0$$

该公式可以进一步简化成如下形式：

$$10 \quad O = (P_0 \bar{s}_1 \bar{s}_0 + P_1 \bar{s}_1 s_0 + P_2 s_1 \bar{s}_0 + P_3 s_1 s_0) \bar{s}_2 + (P_4 \bar{s}_1 \bar{s}_0 + P_5 \bar{s}_1 s_0 + P_6 s_1 \bar{s}_0 + P_7 s_1 s_0) s_2 \\ = [(P_0 \bar{s}_0 + P_1 s_0) \bar{s}_1 + (P_2 \bar{s}_0 + P_3 s_0) s_1] \bar{s}_2 + [(P_4 \bar{s}_0 + P_5 s_0) \bar{s}_1 + (P_6 \bar{s}_0 + P_7 s_0) s_1] s_2$$

通过 MultiplexedAdd MA，计算当前代码长度  $L_i$  和在前代码长度之和  $\sum_{k=0}^{i-1} L_k$  并选择下一代码长度  $L_{i+1}$  是有可能的。通过使用 MultiplexedAdd MA，两个代码长度之间的延迟由  $\log(N)$  位加法器和多路复用器组的延迟降低到  $\log(N)$  全加器和 2-1 多路复用器的延迟。以相应的形式，通过使用 3-4 与或门和 2-2 与或门以及反相器，而将逻辑级由  $2\log(N)$  级减少到  $\log(N)+1$  级。图 5 表示根据本发明的第二个优选实施例，通过 MultiplexedAdd MA 发现代码字的系统。  
15

根据本发明的可变长编码解码器可以作为电子装置的一部分实现，例如，作为单独的逻辑电路，借助在数字信号处理器中的专用集成电路（ASIC），或  
20 处理器中的功能单元。代表性地，上述电子装置也包括其他功能，例如向用户显示已解码的信息的装置以及控制电子装置的处理器。

很明显的是本发明并不局限于上述实施例，而是能在附加的权利要求的范围内加以修改。

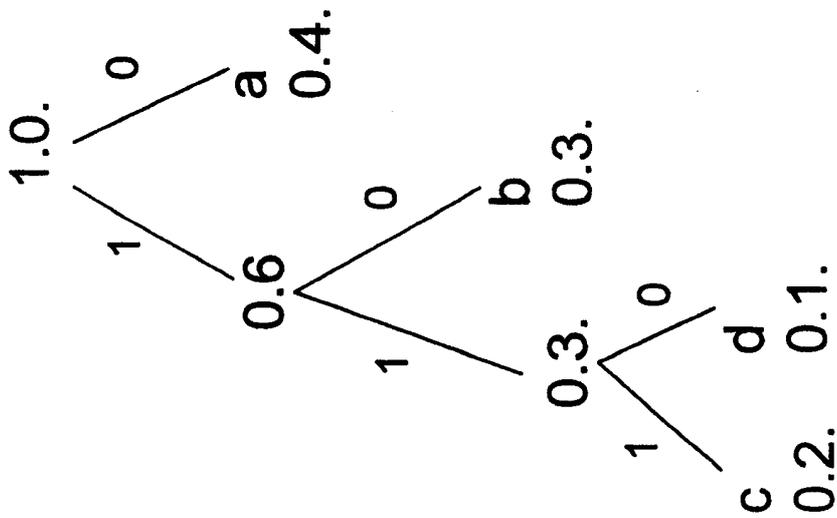


图 1a

符号	H(x)	P(x)	代码
a	1.32	0.4	0
b	1.74	0.3	10
c	2.32	0.2	111
d	3.32	0.1	110

图 1b

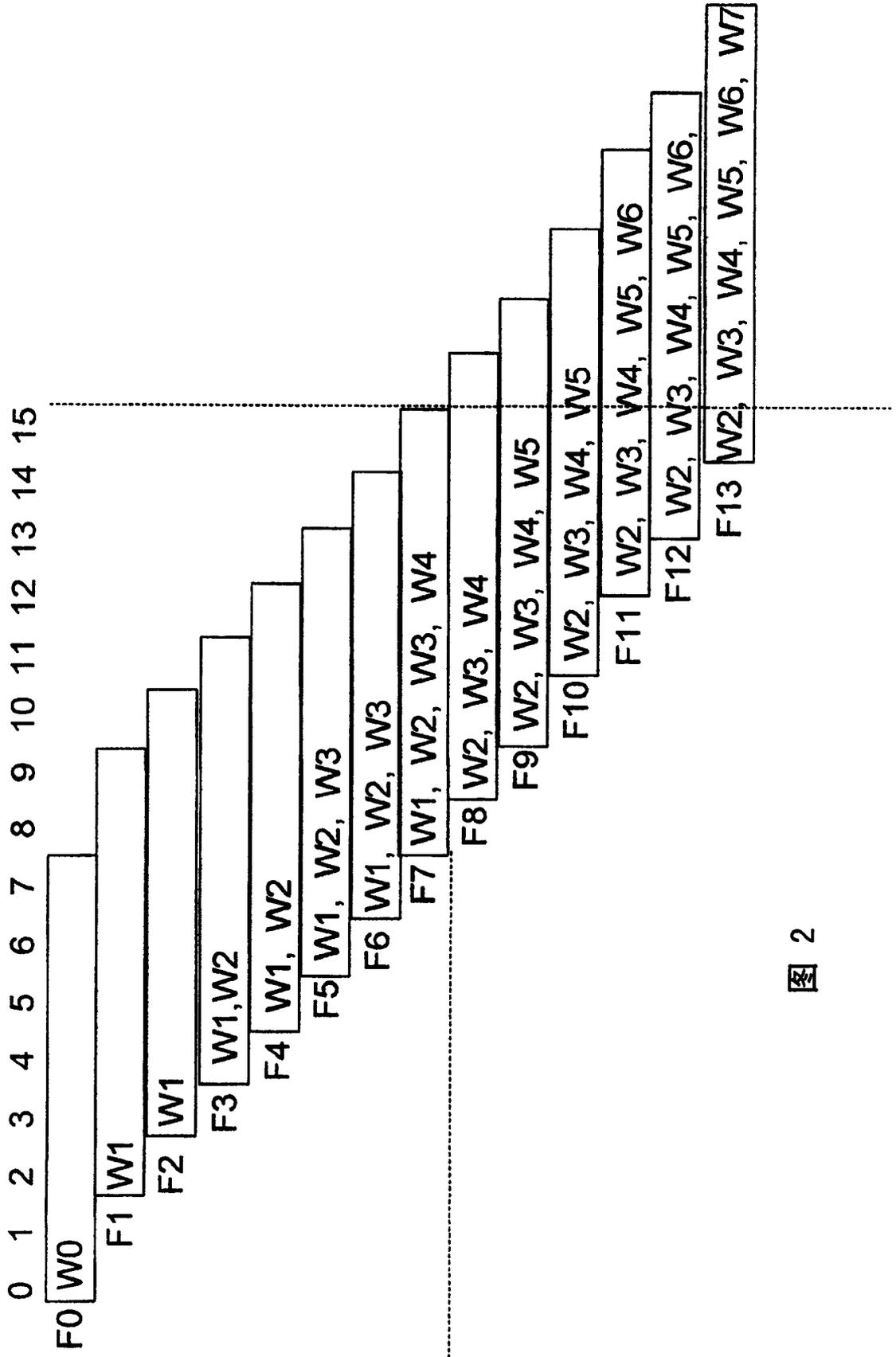


图 2

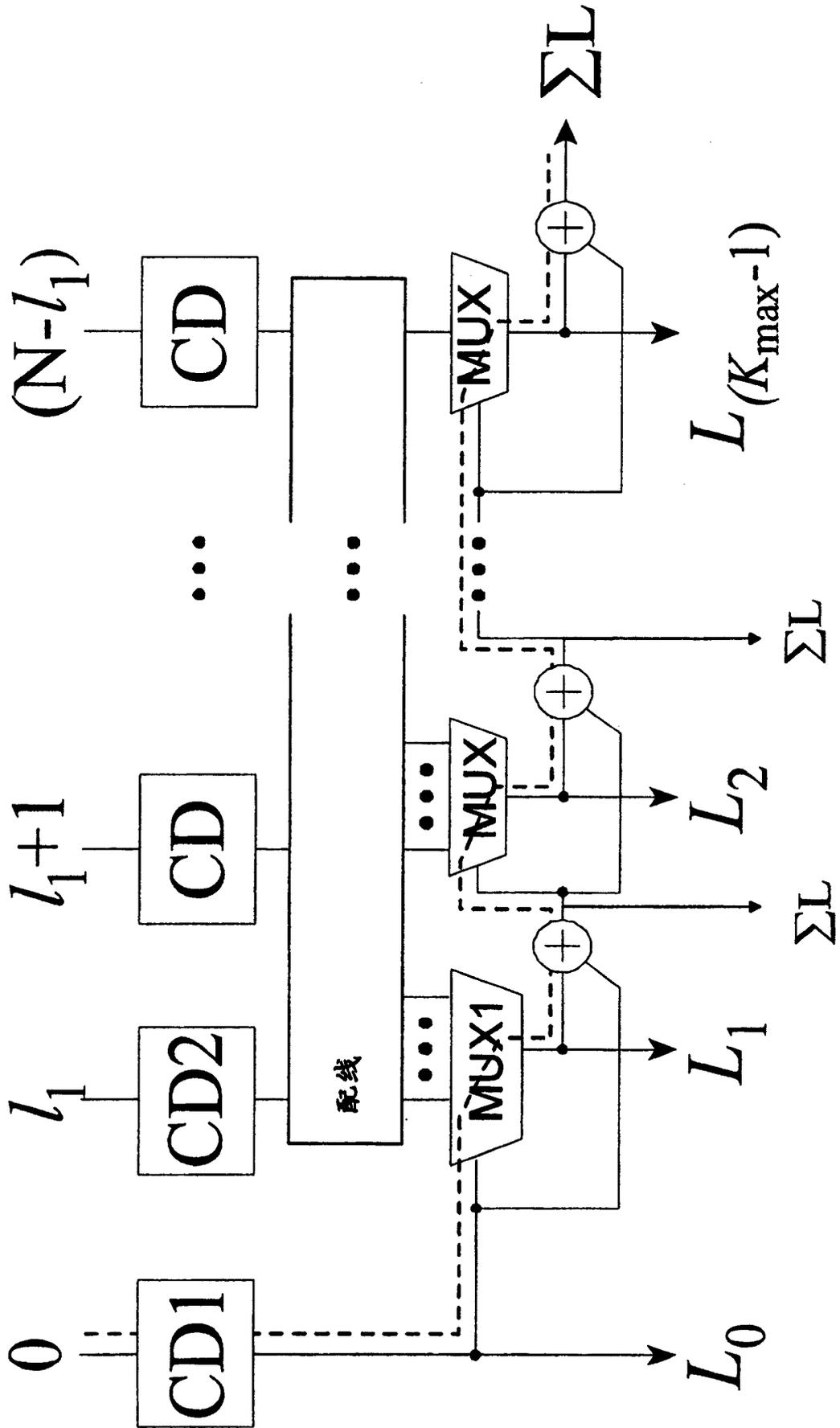


图 3

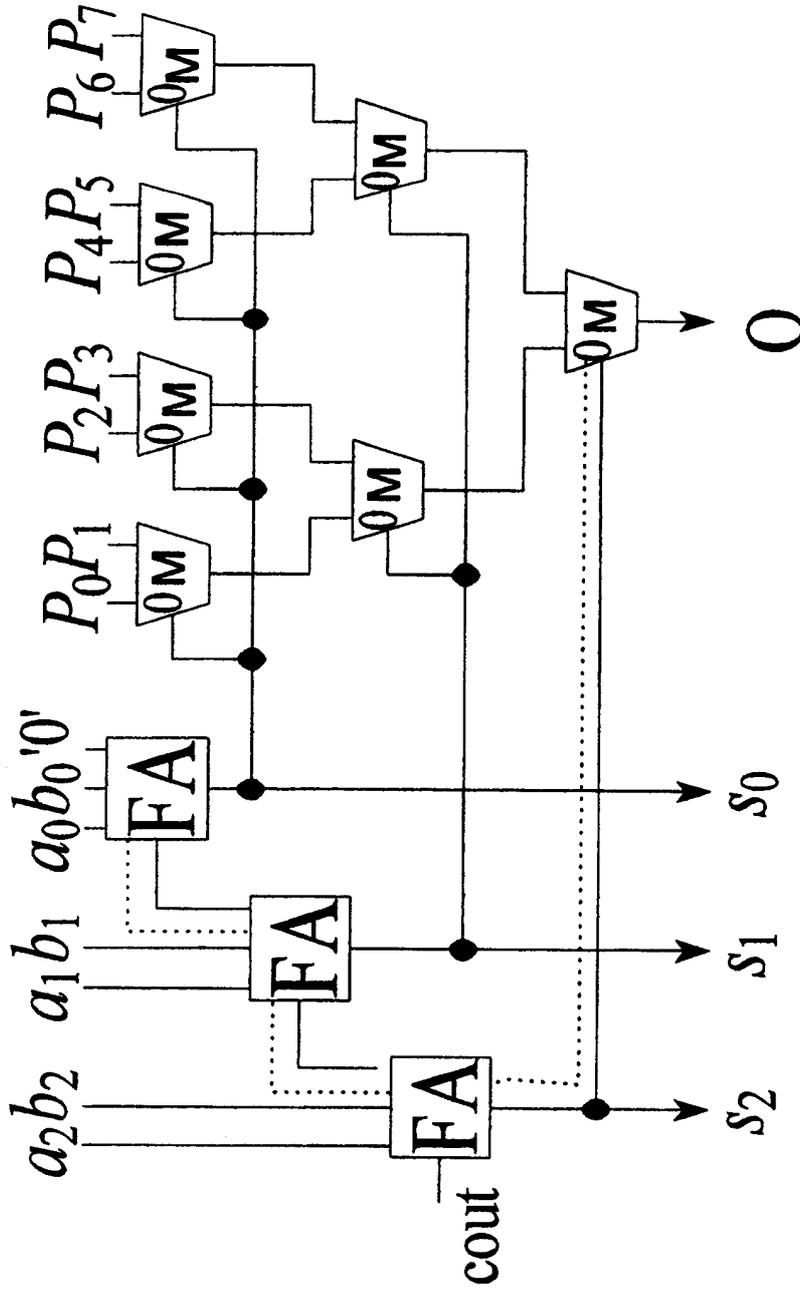


图 4b

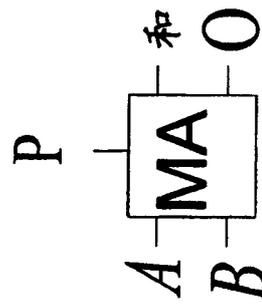


图 4a

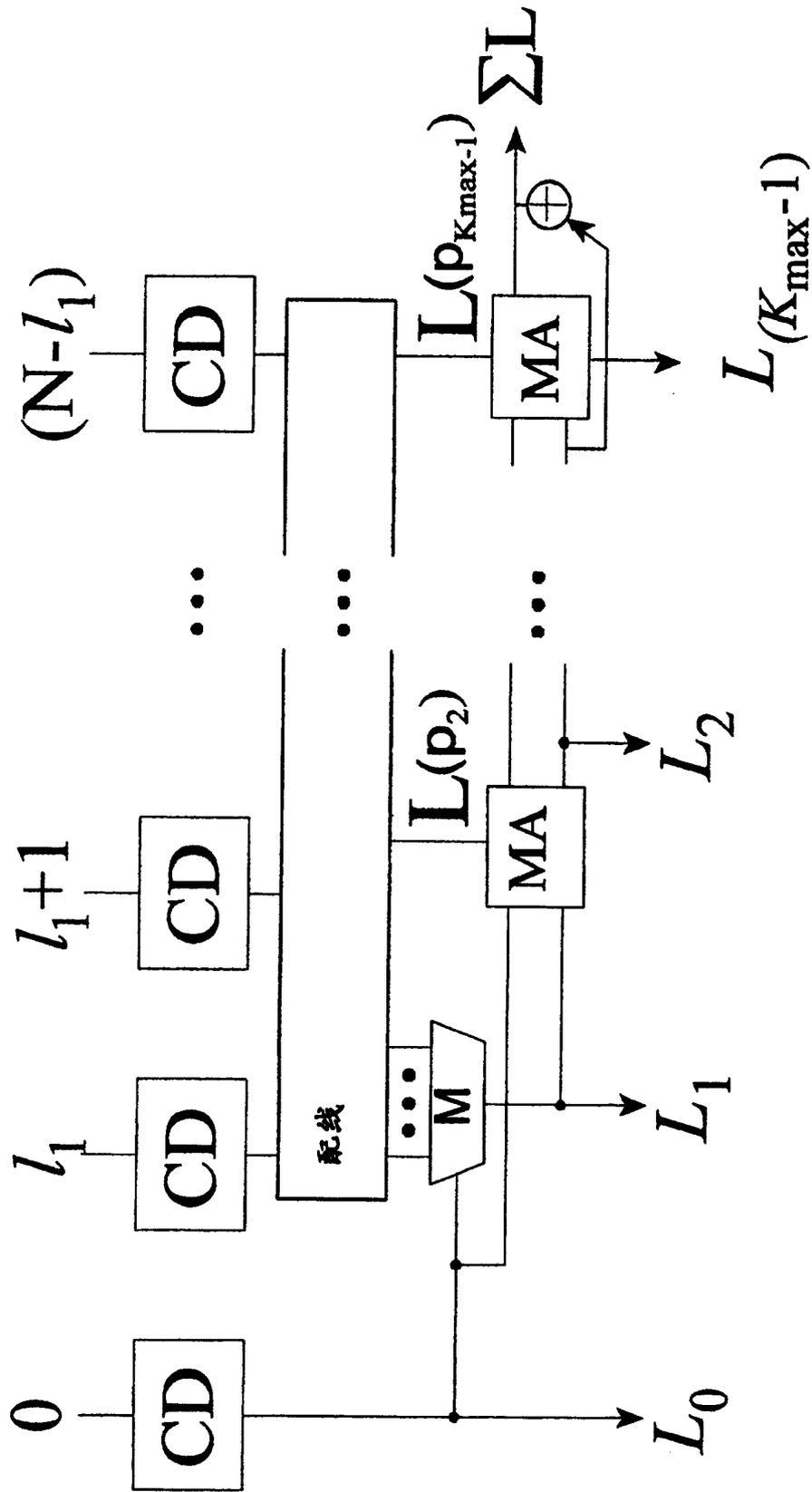


图 5