[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04J 13/00
H03M 13/23

[12] 发明专利说明书

[21] ZL 专利号 00118728.7

[45] 授权公告日 2003 年 11 月 26 日

[11] 授权公告号 CN 1129257C

[22] 申请日 2000.6.20 [21] 申请号 00118728.7

[71] 专利权人 华为技术有限公司
地址 518057 广东省深圳市科技园科发路华为用户服务中心大厦
[72] 发明人 苏 宁
审查员 耿晓芳

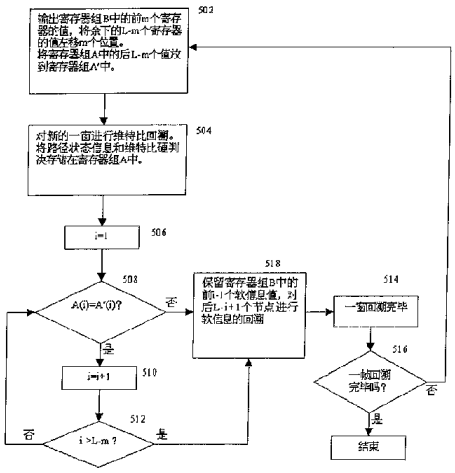
[74] 专利代理机构 上海专利商标事务所
代理人 钱慰民

权利要求书 4 页 说明书 12 页 附图 5 页

[54] 发明名称 串行回溯的最大似然解码方法及其使用该方法的解码器

[57] 摘要

揭示了一种串行回溯的最大似然解码方法，该方法将前一窗幸存路径上各节点的状态号与后一窗幸存路径上各节点的状态号进行比较，以此判断路径的衔接性。 如果路径完全衔接，那么只对后一窗新接收的节点进行并行路径软回溯，并且保留寄存器组中前一窗的软信息。 如果路径部分衔接，那么保留寄存器组中与路径衔接部分对应的软信息，并对非衔接部分的节点进行并行路径回溯。如果路径完全不衔接，那么需要对后一窗中所有节点进行并行路径软回溯。



ISSN1008-4274

1. 一种串行回溯的最大似然解码方法，其特征在于，所述方法包括以下步骤：

(a) 对长度为L的一窗比特流进行维特比回溯，获得幸存路径，并存储每个节点的维特比硬判决以及各节点在所述幸存路径上的路径状态信息；

(b) 根据步骤(a)获得的所述维特比硬判决，从所述窗的最后一个节点开始对所述窗中所有节点作并行路径回溯，并存储所述所有节点的软信息和并行路径硬判决；

(c)对步骤（b）获得的所述软信息和所述硬判决进行m点输出，其中m是$1 \leq m \leq L$的整数；

(d) 用步骤(c)输出的所述硬判决对相应的所述软信息进行调制，获得软输出；

(e)将所述窗口移动m个节点，并接收m个新数据，从而形成一个新的窗口；

(f)对所述新的窗口进行维特比回溯，获得新的幸存路径，并存储相应的维特比硬判决以及所有节点在所述新的幸存路径上的路径状态信息；

(g) 比较新的幸存路径与前一窗的幸存路径，判断所述新的幸存路径与前一窗的幸存路径是否衔接；

(h) 根据步骤(g)的判断结果，并根据所述新的维特比硬判决，有选择地对所述新窗中的节点进行并行路径回溯，获得相应的软信息和并行路径硬判决，并且；

(i) 对步骤(h)获得的所述软信息和所述硬判决进行m点输出；

(j) 用步骤（i）输出的所述硬判决对相应的所述软信息进行调制，获得软输出；以及

(k) 重复步骤（e）至步骤（j）的过程，直到对一帧数据处理完毕。

2. 如权利要求1所述的最大似然解码方法，其特征在于，步骤(h)有选择地进行并行路径回溯至少包括对步骤(e)接收的m个节点进行并行路径回溯。

3. 如权利要求2所述的最大似然解码方法，其特征在于，所述步骤(g)判定所述新窗的幸存路径与前一窗的幸存路径完全不衔接，并且所述步骤(h)包括对所述新

窗中的所有节点进行并行路径的回溯，并更新前一窗获得的所有软信息和并行路径硬判决。

4. 如权利要求2所述的最大似然解码方法，其特征在于，所述步骤(g)判定所述新窗的幸存路径与前一窗的幸存路径部分衔接，并且所述步骤(h)包括保留前一窗获得的与幸存路径衔接部分对应的软信息和并行路径硬判决，对所述新窗中幸存路径与前一窗不衔接的部分中的节点进行并行路径回溯，以及用获得的软信息和并行路径硬判决更新前一窗中与非衔接部分对应的软信息和并行路径硬判决。

5. 如权利要求2所述的最大似然解码方法，其特征在于，所述步骤(g)判定所述新窗的幸存路径与前一窗的幸存路径完全衔接，并且所述步骤(h)包括对所述新窗新接收的m个节点进行并行路径回溯。

6. 如权利要求1所述的最大似然解码方法，其特征在于，所述步骤(b)和步骤(h)包括当回溯经过某个节点k时，将该节点在幸存路径上的硬判决与并行路径上的硬判决进行比较，如果不相等，则作以下计算：

$$llr=\min(llr', Mdiff_n)$$

其中，llr是当前更新后的软信息值，llr′是前一时刻保留的软信息值，而$Mdiff_n$是回溯起点n处幸存路径上硬判决S_n之累计路径度量值与判决$1-S_n$之累计路径度量值的差。

7. 如权利要求1所述的最大似然解码方法，其特征在于，m=1。

8. 如权利要求1所述的最大似然解码方法，其特征在于，m>1。

9. 一种使用串行回溯最大似然解码方法的解码器，它包括：

分支度量计算器，用于计算从某个节点的某个状态到下个节点的某个状态的分支路径度量值；

加比选计算器，用于计算所述分支路径度量值与先前路径上的累计路径度量值之和，计算出当前状态的两个累计路径度量值及其差的绝对值，对计算得到的两个累计路径度量值进行比较，从中选择较大的累计路径度量值及其相应的路径；

路径存储器，用于存储所述加比选计算器选择的所述路径；

状态度量存储器，用于存储所述加比选计算器选择的所述累计路径度量值；

差值存储器，用于存储所述加比选计算器计算得到的当前状态的两个累计路径度量值之差的绝对值；

回溯处理器，它包括：

回溯处理单元，用于根据来自路径存储器的路径信息以及来自差值存储器的累计路径度量值之差，进行维特比回溯运算以及并行路径回溯运算；

第一寄存器组，长度为L，用于存储所述回溯处理单元对当前窗进行维特比回溯所获得的路径信息和硬判决；

第二寄存器组，长度为L，用于存储所述回溯处理单元对当前窗进行并行路径回溯所获得的软信息和硬判决；

符号调制电路，用于对所述回溯处理器输出的软信息进行硬判决调制，输出软输出；以及

控制电路，它与上述各单元相连，用于控制这些单元之间的联络；

其特征在于，所述回溯处理器还包括：

第三寄存器组，长度为L-m，用于存储前一窗中所述第一存储器组中后L-m个寄存器所存储的路径信息，其中m表示一窗回溯完毕时从第二寄存器组中输出软信息的节点数；

比较单元，用于按顺序比较所述第一寄存器组和所述第三寄存器组中相应寄存器中的路径信息，以判断当前窗的幸存路径是否与前一窗的幸存路径衔接，并向所述控制电路发送一信号；

计数器，用于对回溯节点进行计数；

并且，所述控制器被构造成，根据来自所述比较单元的信号，命令所述回溯处理单元对当前窗中的节点有选择地进行并行路径回溯，并且当所述计数器中的计数超过一预定值时，命令输出第二寄存器组中的前 m 个寄存器所存储的软信息值和并行路径硬判决，对第二寄存器组左移 m 位，并且将第一寄存器组中前 L-m 个寄存器中的路径信息存储到第三寄存器组中。

10. 如权利要求 9 所述的解码器，其特征在于，所述回溯处理单元被构造成用于至少对 m 个节点进行并行路径回溯。

11. 如权利要求10所述的解码器，其特征在于，所述比较单元发送的所述信号表示所述当前窗的幸存路径与前一窗的幸存路径完全不衔接，并且所述回溯处理单元被

构造成用于对所述当前窗中的所有节点进行并行路径回溯，并更新所述第二寄存器组中的所有软信息和并行路径硬判决。

12. 如权利要求10所述的解码器，其特征在于，所述比较单元发送的所述信号表示所述当前窗的幸存路径与前一窗的幸存路径部分衔接，并且所述回溯处理单元被构造成用于对所述当前窗中幸存路径与前一窗不衔接的部分中的节点进行并行路径回溯，并更新所述第二寄存器组中与非衔接部分对应的软信息和并行路径硬判决。

13. 如权利要求10所述的解码器，其特征在于，所述比较单元发送的所述信号表示所述当前窗的幸存路径与前一窗的幸存路径完全衔接，并且所述回溯处理单元被构造成用于对所述当前窗接收的m个节点进行并行路径回溯。

14. 如权利要求9所述的解码器，其特征在于，$m=1$。

15. 如权利要求 9 所述的解码器，其特征在于，$1<m\leq L$。

串行回溯的最大似然解码方法及其使用该方法的解码器

本发明涉及无线移动通信移动系统中的特博码（Turbo）的解码方法，尤其涉及一种基于软输入软输出的维特比算法的解码方法，即最大似然解码方法， 以及使用该解码方法的解码器。

在无线通信系统中，传输信号会因传输介质不均匀和不稳定而受到时间扩散、衰落等因素的干扰，致使接收到的比特发生随机性的差错。为了防止信道噪声的干扰，必须采用一定的方式来提高信息的传送可靠性和有效性。实践证明，通过增加冗余度来降低误码率的纠错编码方法是一类很有效的手段。尤其，在移动通信和卫星通信系统中，纠错编码方法得到广泛的应用。

Turbo码是一种纠错能力很强的码。其编码器可以由两个或多个子编码器通过串联或并联的级联方式构成，较普遍的是使两个卷积码编码器并联。

图1是一种Turbo码编码器的结构示意图。它是cdma 2000和WCDMA提案中建议的Turbo码编码器。在该图中，Turbo码编码器10包括两个上下并联的回归系统卷积码（RSC）子编码器14和16。输入信息位一路直接输入第一子编码器14，另一路通过交织器12，输入第二子编码器16。交织器的作用是对输入数据重新排序，调整权重的分布。因此，输入第二子编码器16的比特流的权重分布与第一子编码器14的不同。第一子编码器14和第二子编码器16分别对数据编码，然后将经编码的数据输入打孔器18。打孔器18对两个子编码器14和16输出的多路比特打孔抽样和并串转换，将数据调制到合适的码率（诸如1/2、1/3、或1/4码率等）输出。

图2例示了图1中子编码器14和16的结构。卷积码编码器通常用（n0，k0，m）来表征。其中n0是编码器的输出比特；k0是编码器的输入比特；而m为编码器中移位寄存器的个数。编码器还可以用约束长度K来表示其特性，它等于卷积码编码器内部移位寄存器的个数m加1，用来确定区段信息比特影响的范围。图2示出的是一个（3，1，3）卷积码编码器。其约束长度为4，码率为1/3。

它是cdma2000提案中的结构。如果采用WCDMA提案中的结构，那么编码器为（2，1，3），没有Y_1输出。

由图2可见，编码器包括三个相互串联连接的移位寄存器20。每当输入端输入一位时，各移位寄存器中的内容依次向右传递。编码器还包括多个模2加法器22，它们按照一定的编码规则对输入信号和各级移位寄存器的输出信号作加法处理。在该图中，对应于一个比特的信息输入，编码器将输出三个比特，即X，Y_0和Y_1。X是与输入信息相同的信息位，Y_0和Y_1是两个校验比特。当X因信道干扰而发生误码时，Y_0和Y_1可用来纠错。另外，编码器还包括一个尾比特控制器24。当一帧数据输入完毕时，需要对移位寄存器20清零。这时，可以将尾比特控制器24的开关切换到下方，通过三个节拍，将三个移位寄存器20内的比特作为输入依次清零。

Turbo码的解码采用递归迭代方式。根据不同的译码算法，主要分为最大后验概率（MAP）译码算法和最大似然（SOVA）译码算法。图3例示了一种Turbo码解码器30的结构。它使用SOVA译码算法。首先，解打孔装置31对接收信号解打孔，它相当于图1所示Turbo编码器10中打孔器18的逆操作。例如，对于（3，1，3）子编码器的情况，解打孔装置31要对接收信息进行串并转换，并通过对打孔器18打掉的信息位补零，将三路信息恢复成六路。在解打孔装置31输出的信号中，对应于第一子编码器14之编码结果的三路信息X、Y_0和Y_1输入第一软输入软输出（SISO）解码器32，对应于第二子编码器16之编码结果的三路信息X′、Y_0'和Y_1'输入第二软输入软输出解码器。除此之外，为了提高SISO解码器的解码精度，每个解码器32和33还需要输入一个先验信息Z或Z′。先验信息Z的初始值可以设置为零。具体地说，第一SISO解码器32对第一子编码器14的编码结果解码，除输出软信息之外，还输出一附加的外赋信息。这些输出信息经交织器34交织后，作为先验信息Z′输入第二SISO解码器33。第二SISO解码器33对第二子编码器16的编码结果解码，输出相应的软信息和外赋信息。然后，这些输出信息经解交织器36解交织，还原到交织前的顺序，并作为先验信息Z输入第一SISO解码器32。如此反复迭代，解码精度越来越高，误码率越来越低。经过多次迭代后，如果认为达到了精度要求，则输入解交织器37进行解交织，还原交织前的顺序。由于解交织器的输出是一些表示概率的带

符号的数（例如，0.8、-1.2、5.5等等），所以需要用判决器38对解交织后的信息作硬判决。当信息大于0时，判决器输出1；当信息小于0时，判决器输出0。经解码后得到的信息不会等于0。最后，判决器输出解码结果，恢复原来的信息X。

美国专利第5,406,570号（下称专利'570号）介绍了一种软输出维特比算法（即，SOVA）解码器的结构（对应于图3中SISO 32，33）。该专利的发明名称为“通过判决加权对卷积码进行最大似然解码的方法，及相应的解码器（Method for a Maximum Likelihood Decoding of a Convolutional Code with Decision Weighting, and Corresponding Decoder）”。其发明内容通过引用包括在此。

专利'570号先对长度为L的第一个网格图和长度为L'的第二个网格图进行维特比回溯（即，幸存路径回溯）。然后，在第二网格图内，从L节点开始寻找从L节点至L＋L'节点的并行回溯路径。在第二个网格图内，对于某节点k，当幸存路径的硬判决S_k和并行路径的硬判决S_k'不相等时，做如下计算：

$$llr=\min(llr',\ Mdiff_k) \qquad (1)$$

其中，llr是当前更新后的软信息值，llr'是前一时刻保留的软信息值，而$Mdiff_k$是节点k处幸存路径上硬判决S_k之累计路径度量值与判决$1-S_k$之累计路径度量值的差。

专利'570号采用的是串行回溯法。从严格意义上讲，串行回溯法是对一窗中的每个节点依次按公式(1)进行比较，然后将各节点的软信息结果存放在移位寄存器组内。当对一窗中的所有节点（例如，一窗包括L个节点）回溯结束后，窗口滑动一个节点，输出移位寄存器组内第一个寄存器所存储的数据，并接收一个新的数据。然后，再对新窗中的每个节点重复上述比较过程。如此一次次滑动窗口，最后完成对一帧比特流的软信息回溯。

然而，在专利'570号中，当滑动窗口时，只对第二个网格图中新增加的一个节点进行并行路径回溯（即，窗口滑动一个节点，只进行一次并行路径回溯）。这意味着，发明人假设当窗口滑动一个节点并由此输出一个老数据和接收一个新数据时，窗口滑动前后第一网格图和第二网格图里的幸存路径没有改变。显然，这种假设只有在一窗长度等于帧长时才能保证100%的正确。

或者说，只有在回溯窗长度比较大并且信道情况比较好的情况下，回溯结果才有比较高的可靠性。但是，实际情况往往无法满足这两点。

应该看到，与严格的串行回溯法相比，专利'570号的串行回溯法运算量很小。例如，假设窗口长度为L。按所述严格的串行回溯法，整窗回溯需要进行1+2+…+L=L(L+1)/2次节点回溯。而在专利'570号中，由于假设窗口滑动前后幸存路径不变，所以每次窗口滑动后只要对新接收的一个数据回溯就可以了，即只要进行L次节点回溯。

基于上述分析，我们知道在SOVA算法中，为了保证精度，需要保留比较长的回溯窗口，但这会造成存储器规模过大。反之，如果为了节约存储器而减小回溯窗口长度，则会造成精度下降。

本发明的一个目的是，提供一种即能保证解码精度，而能减小软信息回溯次数的串行回溯 SOVA 解码方法。

本发明的另一个目的是，提供一种使用本发明串行回溯 SOVA 解码方法的解码器。

为了达到上述目的，依照本发明的一个方面，提供了一种串行回溯的最大似然解码方法，该方法包括以下步骤：

(a) 对长度为L的一窗比特流进行维特比回溯，获得幸存路径，并存储每个节点的维特比硬判决以及各节点在所述幸存路径上的路径状态信息；

(b) 根据步骤(a)获得的所述维特比硬判决，从所述窗的最后一个节点开始对所述窗中所有节点作并行路径回溯，并存储所述所有节点的软信息和并行路径硬判决；

(c)对步骤（b）获得的所述软信息和所述硬判决进行m点输出，其中m是1≤m≤L的整数；

(d) 用步骤(c)输出的所述硬判决对相应的所述软信息进行调制，获得软输出；

(e)将所述窗口移动m个节点，并接收m个新数据，从而形成一个新的窗口；

(f)对所述新的窗口进行维特比回溯，获得新的幸存路径，并存储相应的维特比硬判决以及所有节点在所述新的幸存路径上的路径状态信息；

(g) 比较新的幸存路径与前一窗的幸存路径，判断所述新的幸存路径与

前一窗的幸存路径是否衔接；

(h) 根据步骤(g)的判断结果，并根据所述新的维特比硬判决，有选择地对所述新窗中的节点进行并行路径回溯，获得相应的软信息和并行路径硬判决；

(i) 对步骤(h)获得的所述软信息和所述硬判决进行m点输出；

(j) 用步骤（i）输出的所述硬判决对相应的所述软信息进行调制，获得软输出；以及

(k) 重复步骤（e）至步骤（j）的过程，直到对一帧数据处理完毕。

在上述本发明的最大似然解码方法中，至少要对新接收的m个节点进行并行路径回溯。并且，如果判定所述新窗的幸存路径与前一窗的幸存路径完全不衔接，那么对所述新窗中的所有节点进行并行路径的回溯。如果判定所述新窗的幸存路径与前一窗的幸存路径部分衔接，那么对所述新窗中幸存路径与前一窗不衔接的部分中的节点进行并行路径回溯。如果判定所述新窗的幸存路径与前一窗的幸存路径完全衔接，那么对所述新窗新接收的m个节点进行并行路径回溯。

在上述方法中，并行路径回溯所得到的软信息和硬判决即可以采用单点输出，也可以采用多点输出。

依照本发明的另一方面，提供了一种使用上述串行回溯最大似然解码方法的解码器，它包括：

分支度量计算器，用于计算从某个节点的某个状态到下个节点的某个状态的分支路径度量值；

加比选计算器，用于计算所述分支路径度量值与先前路径上的累计路径度量值之和，计算出当前状态的两个累计路径度量值及其差的绝对值，对计算得到的两个累计路径度量值进行比较，从中选择较大的累计路径度量值及其相应的路径；

路径存储器，用于存储所述加比选计算器选择的所述路径；

状态度量存储器，用于存储所述加比选计算器选择的所述累计路径度量值；

差值存储器，用于存储所述加比选计算器计算得到的当前状态的两个累计路径度量值之差的绝对值；

回溯处理器，它包括：

回溯处理单元，用于根据来自路径存储器的路径信息以及来自差值存储器的累计路径度量值之差，进行维特比回溯运算以及并行路径回溯运算；

第一寄存器组，长度为L，用于存储所述回溯处理单元对当前窗进行维特比回溯所获得的路径信息和硬判决；

第二寄存器组，长度为L，用于存储所述回溯处理单元对当前窗进行并行路径回溯所获得的软信息和硬判决；

符号调制电路，用于对所述回溯处理器输出的软信息进行硬判决调制，输出软输出；以及

控制电路，它与上述各单元相连，用于控制这些单元之间的联络；

另外，所述回溯处理器还包括：

第三寄存器组，长度为L-m，用于存储前一窗中所述第一存储器组中后L-m个寄存器所存储的路径信息，其中m表示一窗回溯完毕时从第二寄存器组中输出软信息的节点数；

比较单元，用于按顺序比较所述第一寄存器组和所述第三寄存器组中相应寄存器中的路径信息，以判断当前窗的幸存路径是否与前一窗的幸存路径衔接，并向所述控制电路发送一信号；

计数器，用于对回溯节点进行计数；

并且，所述控制器被构造成，根据来自所述比较单元的信号，命令所述回溯处理单元对当前窗中的节点有选择地进行并行路径回溯，并且当所述计数器中的计数超过一预定值时，命令输出第二寄存器组中的前 m 个寄存器所存储的软信息值和并行路径硬判决，对第二寄存器组左移 m 位，并且将第一寄存器组中前 L-m 个寄存器中的路径信息存储到第三寄存器组中。

由此可见，本发明用一附加的寄存器组存储对前一窗数据进行维特比回溯所获得的路径状态号。然后，在对后一窗回溯时，将前一窗幸存路径上各节点的状态号与后一窗幸存路径上各节点的状态号进行比较，以此判断路径的衔接性。如果路径完全衔接，那么只对后一窗新接收的节点进行并行路径软回溯，并且保留寄存器组中前一窗的软回溯信息和并行路径硬判决。如果路径部分衔接，那么保留寄存器组中与衔接部分对应的节点的软信息和硬判决，而对路径非衔接部分的

节点（包括新接收到的节点）进行并行路径回溯，用回溯得到软信息和硬判决更新寄存器组相应寄存器的内容。如果路径完全不衔接，那么需要对后一窗中所有节点进行并行路径软回溯，并更新寄存器组中所有寄存器的内容。

由此可见，本发明通过保留寄存器组中的路径状态信息来判断窗口滑动前后路径的衔接性，并根据判断结果有选择地对新窗中的节点进行并行路径回溯。因此，避免了对幸存路径相同的部分进行软回溯，从而减少了回溯次数。另一方面，在本发明中，由于软信息回溯时参照的幸存路径与实际的幸存路径没有偏差，所以保证了解码精度。

图1是一种Turbo码编码器的结构；

图2是Turbo码编码器中RSC子编码器的结构图；

图3是Turbo码解码器的结构图；

图4是一示意图，例示了依照本发明较佳实施例的SOVA解码方法的工作原理；

图5是一流程图，例示了依照本发明较佳实施例的SOVA解码方法的工作过程；

图 6 一方框图，例示了图 3 所示 Turbo 码解码器中软输入软输出解码器的结构；

图 7 是一方框图，例示了回溯处理器的结构。

以下结合图 4 和图 5，描述本发明的较佳实施例。

图 4 是一示意图，例示了依照本发明较佳实施例的 SOVA 解码方法的工作原理。在图 4 中，虚线表示幸存路径，实线表示并行路径。如图 4 所示，假设在进行幸存路径的维特比回溯和并行路径的软回溯时，一窗包括 L 个节点。先对第一窗进行维特比回溯，获得相应的幸存路径，并将每个节点的维特比硬判决 S_k（S_k =0 或 1，0≤k≤L-1）以及每个节点在幸存路径上的状态号 S[k]（例如 S[k]=0～7，0≤k≤L-1）分别存储在寄存器组 A 的相应寄存器中。寄存器组 A 可以由 L 个寄存器组成，每个寄存器包括两个域，分别用于存储相应节点的维特比硬判决和状态号。然后，从最后一个节点 L-1 开始对第一窗的所有节点做并行路径的软信息回溯。具体地说，在对节点 n（0≤n≤L-1）进行并行路径回溯，并且回溯经过节点 k（0≤k≤n）时，将节点 k 的并行路径硬判决 S_k'与幸存路径上该节点的硬判决 S_k（即，寄存器组 A 中第 k+1 个寄存器的值）比较。如果不同，则将本次回溯起点 n 的 $Mdiff_n$ 值与寄存器组 B 中第 k+1 个寄存器的值

进行比较。这里，$Mdiff_n$ 值是从节点 n 到前一个节点之两个状态的两个累计度量值之差的绝对值，寄存器组 B 由 L 个寄存器组成，并且每个寄存器包括两个域，分别用于存储并行路径回溯所获得的软信息和硬判决，寄存器的初始值设置为无穷大。经过比较，如果 $Mdiff_n$ 较小，则用 $Mdiff_n$ 更新第 k+1 个寄存器的值。否则保留原来的值。实质上，这种比较和更新过程就是背景技术中给出的公式 1。当对节点 n 一直回溯到第一窗的起始点后，再对节点 n-1 进行并行路径回溯。如此往复，直至第一窗回溯完毕。这时，寄存器组 A 中存储着第一窗 L 个节点的维特比硬判决和相应的状态号，而寄存器组 B 中存储着第一窗 L 个节点的软信息。另外，寄存器组 B 中还存储了与各软信息对应的并行路径硬判决。

一窗回溯完毕后可以单点输出，也可以多点输出。在本实施例中，采用 2 点输出。具体地说，在第一窗回溯完毕后，输出寄存器组 B 前 2 个寄存器中保存的软信息和硬判决。然后，将寄存器组 B 的内容左移两个位置，并将最后二个寄存器的值设为无穷大。另外，还要将寄存器组 A 中后 L-2 个寄存器所存储的路径状态号按顺序存储到一附加寄存器组 A′中。在本实施例中，由于采用 2 点输出，所以寄存器组 A′由 L-2 个寄存器组成。

接下来描述第二窗的回溯。滑动窗口，接收 2 点新的数据。然后，对新的一窗进行维特比回溯，获得新的幸存路径，并将相应的各节点的硬判决和状态号存储在寄存器组 A 中。由图 4 可见，在本实施例中，第二窗维特比回溯终点的状态号与第一窗中相应节点的状态号不同。这说明幸存路径完全不衔接，需要对第二窗的全部节点进行并行路径软回溯。具体地说，在图 4 中，第二窗的终点是节点 2，其状态号是 1，而第一窗中节点 2 在幸存路径上的状态号是 0。旧窗和新窗中第一个节点的状态号不一致，所以幸存路径完全不衔接。因此，在该情况下需要对第二窗的所有节点进行并行路径软回溯。其回溯过程与上述第一窗并行路径回溯的过程一样。

第二窗回溯完毕后，寄存器组 A 中存储着第二窗每个节点的维特比硬判决以及各节点在幸存路径上的状态号，而寄存器组 B 中存储着第二窗全部回溯获得的软信息和并行路径硬判决。

再次输出寄存器组 B 前 2 个寄存器中保存的软信息和硬判决。然后，将

寄存器组 B 的内容左移两个位置，并将最后二个寄存器的值设为无穷大。同样，还要将寄存器组 A 中后 L-2 个寄存器所存储的路径状态号按顺序存储到寄存器组 A'中。

接下来描述第三窗的回溯过程。滑动窗口，接收 2 点新的数据。然后对第三窗进行维特比回溯，获得新的幸存路径，并将相应的各节点的硬判决和状态号存储在寄存器组 A 中。如图 4 所示，第三窗的幸存路径从节点 4 到节点 L+1 的状态号与第二窗中对应节点的状态号相同。这说明幸存路径完全衔接。这时，可以保留寄存器组 B 中前 L-2 个寄存器的值。只需要对新接收的 2 个数据（它们对应于节点 L+2 和 L+3）计算软回溯信息，并用计算得到的软信息和硬判决更新寄存器组 B 中最后二个寄存器的值。并行路径的软回溯方法与前述相同。第三窗回溯完毕后，寄存器组 A 中存储着第三窗每个节点的维特比硬判决以及各节点在幸存路径上的状态号，而寄存器组 B 中存储着第三窗的软信息和并行路径硬判决。与第一和第二窗的情况一样，输出寄存器组 B 中前 2 个寄存器中的内容。然后，将寄存器组 B 的内容左移两个位置，并将最后二个寄存器的值设为无穷大。同样，还要将寄存器组 A 中后 L-2 个寄存器所存储的路径状态号按顺序存储到寄存器组 A'中。

再看第四窗的回溯情况。如图 4 所示，第四窗的幸存路径从节点 6 到节点 9 的状态号与第三窗中幸存路径上相应节点的状态号相同。但第四窗节点 10 的状态号为 2，第三窗节点 10 的状态号为 3，两者不同。这说明幸存路径部分衔接。这时，可以保留寄存器组 B 中前 4 个寄存器的值，计算从节点 10 到节点 L+5 的软回溯信息，并用计算得到的软信息和硬判决更新寄存器组 B 中第 5 至 L 个寄存器的值。并行路径的软回溯方法与前述相同。第四窗回溯完毕后，寄存器组 A 中存储着第四窗每个节点的维特比硬判决以及各节点在幸存路径上的状态号，而寄存器组 B 中存储着第四窗的软信息和并行路径硬判决。

如此按照上述方式，一次次滑动窗口，对每次获得的新窗口进行路径衔接判断，然后根据判断结果有选择地对窗口中的节点进行并行路径回溯，直到完成对一帧比特流的回溯。

在上述实施例中，对第二窗、第三窗以及第四窗并行路径回溯的描述只

是为了说明路径衔接的三种情况。在实际回溯过程中，这三种情况可以按任意次序出现。

图5是一流程图，例示了依照本发明较佳实施例的SOVA解码方法的工作过程。在本实施例中，回溯窗口长度为L，并采用m点。

假设前一窗已经回溯完毕。在步骤 502，输出寄存器组 B 中前 m 个寄存器存储的软信息和硬判决，将后 L-m 个寄存器中的内容左移 m 个位置，并将寄存器组 B 中最后 m 个寄存器的值设为无穷大；将寄存器组 A 中后 L－m 个存储器所存储的路径状态信息存储到附加寄存器组 A′中。在步骤 504 中，接收 m 个新数据，进行新的一窗的维特比回溯，将计算得到的维特比硬判决和各节点在幸存路径上的状态号存储在寄存器组 A 中。在步骤 506，设定 i=1。在步骤 508，判断寄存器组 A 中第 i 个寄存器所存储的路径状态号是否与寄存器组 A′中第 i 个寄存器所存储的路径状态号相等。如果相等，则过程进至步骤 510。在步骤 510，对 i 增 1。然后，在步骤 512，判断是否已对寄存器组 A′所存储的所有状态号都作了比较。如果已全部比较，表示原有的路径信息、软信息和硬判决都可以保留。过程进至步骤 518，对新接收的 L－i＋1＝m 点进行并行路径回溯。一窗回溯完毕后，如果一帧回溯还未完成，则过程回到步骤 502，进行 m 点输出。如果已完成对一帧数据的回溯，则回溯过程结束。

如果步骤 512 判定没有全部比较完寄存器组 A′中所存储的所有状态号，那么过程返回步骤 508，对寄存器组 A 和 A′中下一个寄存器的值进行比较。如果在步骤 508 中，判断寄存器组 A 中第 i 个寄存器所存储的路径状态号不等于寄存器组 A′中第 i 个寄存器所存储的路径状态号，那么过程进至步骤 518。在步骤 518 中，保留寄存器组 B 中前 i－1 个寄存器中的软信息和硬判决，对后面 L-i＋1 个节点进行软信息回溯，并将回溯得到的软信息和硬判决分别存储在寄存器给后 L-i＋1 个寄存器中。然后，过程进至步骤 514，一窗回溯完毕。

接下来结合图6和图7，举例说明实施本发明SOVA解码方法的装置。图6一方框图，例示了图3所示Turbo码解码器中软输入软输出（SISO）解码器的结构。如该图所示，分支路径度量计算器（BMU）41计算从某个节点的某个状态到达下个节点两种可能状态的路径度量值，并将计算结果送入加比选计算器

（ACS）42。加比选计算器42将分支路径度量值与先前路径上的累计路径度量值相加，计算出当前状态的两个累计路径度量值及其差的绝对值，对计算得到的两个累计路径度量值进行比较，从中选择较大的累计路径度量值及其相应的路径，然后将选择结果分别输入状态度量存储器（SMM）45和路径存储器43，并将当前状态上计算得到的两个累计路径度量值之差的绝对值输入差值存储器46中。回溯处理器47依照根据路径存储器43提供的路径信息以及差值存储器46提供的累计路径度量值之差，进行维特比回溯和并行路径回溯，输出软信息llr和硬判决。符号调制48对回溯处理器47输出的软信息进行硬判决调制，也称符号调制。具体地说，若硬判决输入为1，则对软信息乘正号，若硬判决输入为0，则对软信息乘负号。由此获得软输出。这时，如果解码还未达到预定的精度，则要作归一化49计算，作为下一级迭代输入的外赋信息。

另外，在该软输入软输出解码器中，还包括控制器 44，它用于控制上述各部件之间的联络。

图 7 是一方框图，例示了回溯处理器 47 的结构。如图 7 所示，回溯处理器 47 包括一回溯处理单元 470，它根据来自路径存储器 43 的路径信息以及差值存储器 46 提供的累计路径度量值之差，进行维特比回溯和并行路径回溯。将维特比回溯得到的状态号和维特比硬判决存储在寄存器组 A 中，而将并行路径溯得到的软信息和并行路径硬判决存储在寄存器组 B 中。当一窗回溯完毕后，在控制器 44 的控制下，输出寄存器组 B 中前 m 个寄存器的值，并使后 L-m 个寄存器中的值左移 m 位。另外，控制器 44 还指示将寄存器组 A 中的后 L-m 个寄存器所存储的路径状态号存储到寄存器组 A′中。然后，控制器 44 控制回溯处理单元 470 对新的一窗进行维特比回溯，并将新获得的路径状态号和维特比硬判决存储在寄存器组 A 中。然后，在比较单元 474 中，将寄存器组 A 中关于新一窗的路径状态号与寄存器组 A′中关于前一窗的路径状态号依次进行比较。当发现路径状态号不同时，比较单元 474 向控制器 44 发送一信号。控制器 44 根据该信号决定可以保留寄存器组 B 中哪些寄存器中的值，需要更新哪些寄存器中的值，以及回溯处理单元 470 需要对哪些节点进行并行路径回溯。回溯处理单元 470 根据控制器 44 的指令，对新一窗的节点有选择地进行并行路径回溯。另外，回溯处理器 47 还包括一计数器，控制器根据计

数器的计数判断是否一窗回溯结束。如果结束，则如前所述，命令输出寄存器组 B 中的 m 个值，对寄存器组 B 进行左移操作，以及将寄存器组 A 中的 L-m 值存储到寄存器组 A′中。

由于本发明在对每一窗进行软信息回溯前，要将前一窗幸存路径的路径状态信息与当前窗的路径状态信息进行比较，所以在存储器规模上，与严格的串行回溯法相比，本发明的串行回溯需要增一个附加的寄存器组A′。对于单点输出的情况，也最好增加L-1个寄存器。从运算量来说，即使采用单点输出，本发明对每窗L长度的回溯也最多增加 2(L-1)次比较操作（对应于步骤508和步骤512），和(L-1)次存储操作（对应于步骤502中寄存器组A和A′之间的转存）。如果计算平均概率，则增加了(k-1)×3/2次比较和(k-1)/2次存储操作。但如果在软信息回溯过程中幸存路径完全衔接的情况出现较多，或者幸存路径部分衔接时衔接部分的路径长度比较长，那么可以大大减少软回溯的次数。

应该理解，在本发明方法中，由于所有软信息回溯时参照的幸存路径与实际幸存路径没有偏差，所以软回溯的精度是保证的。

对于数字信号处理（DSP）来说，由于Turbo码解码器串行回溯时软回溯的运算量过大，所以DSP只能适用于Turbo码低码率的情况。但通过使用本发明的方法，DSP可以应用于Turbo码中码率较高情况。同时，在时钟周期比较宽裕而规模有限制的FPGA、PLD设计中，也可以使用本发明的方法。

虽然本发明的较佳实施例描述了 cdma2000 和 WCDMA 提案中的 Turbo 码编码器和其 RSC 子编码器的解码方法，但是本发明中所阐述的思想和算法在其他方式下的衍生变化亦属于本申请发明之权利保护范围之内。

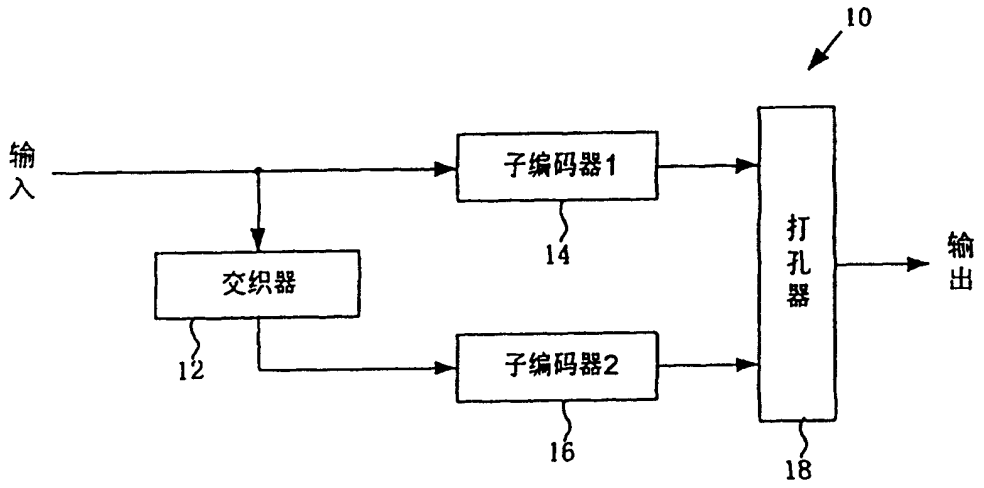

10
输入
子编码器1
14
交织器
12
子编码器2
16
打孔器
18
输出

图 1

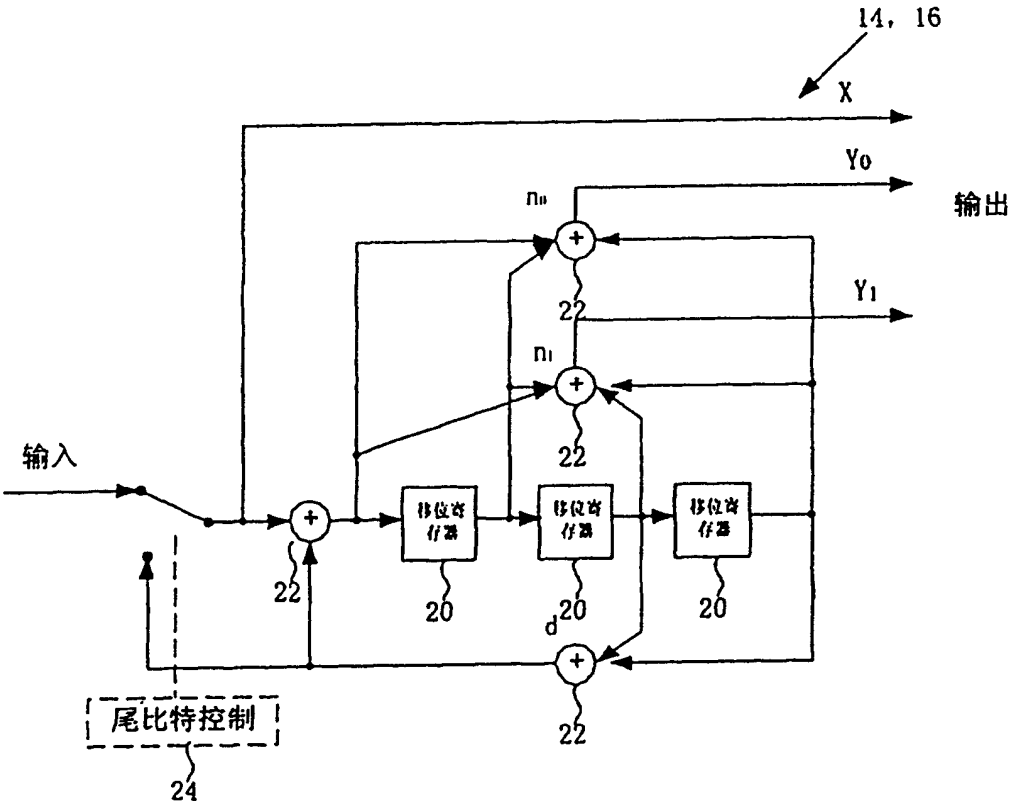

14，16
X
Y0
输出
n0
22
Y1
n1
22
输入
移位寄存器
移位寄存器
移位寄存器
22
20
20
d
20
22
尾比特控制
24

图 2

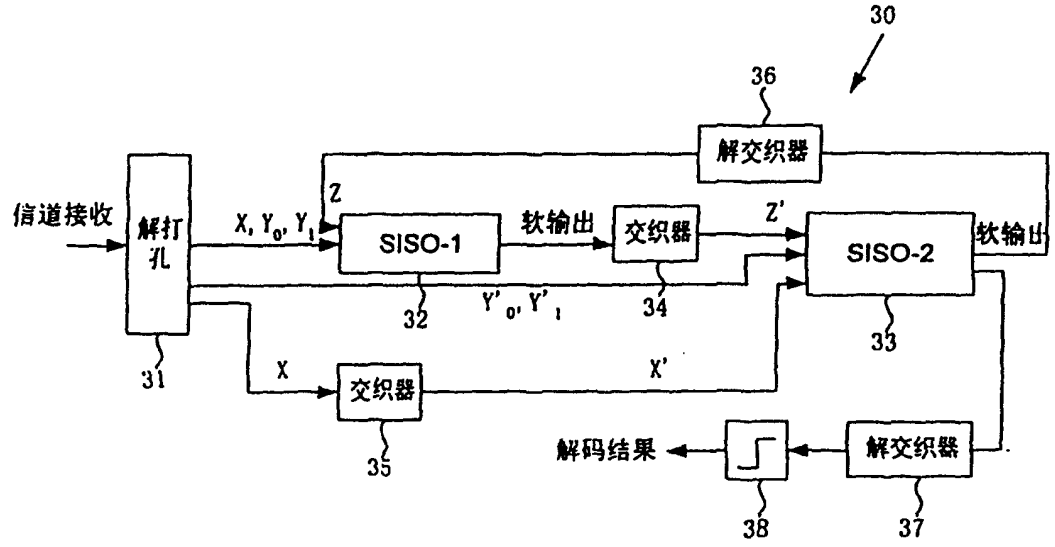

30
36
解交织器
信道接收
解打孔
X, Y0, Y1
Z
SISO-1
软输出
交织器
Z'
SISO-2
软输出
Y'0, Y'1
32
34
33
31
X
交织器
X'
35
解码结果
解交织器
38
37

图 3

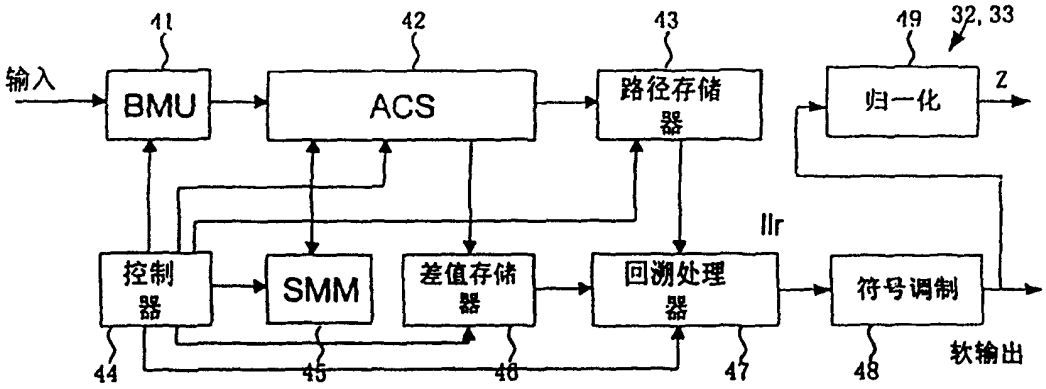
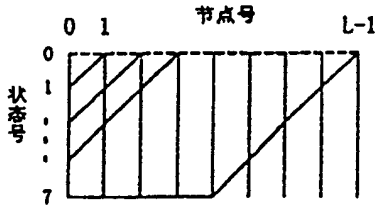
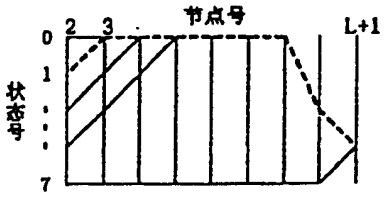

41
42
43
19
32, 33
输入
BMU
ACS
路径存储器
归一化
Z
llr
控制器
SMM
差值存储器
回溯处理器
符号调制
软输出
44
45
46
47
48

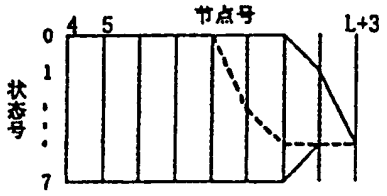
图 6



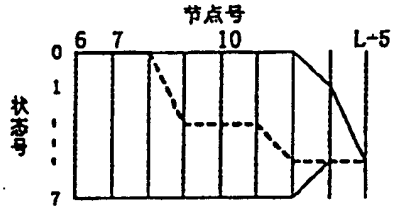
第一窗



第二窗



第三窗



第四窗

图 4

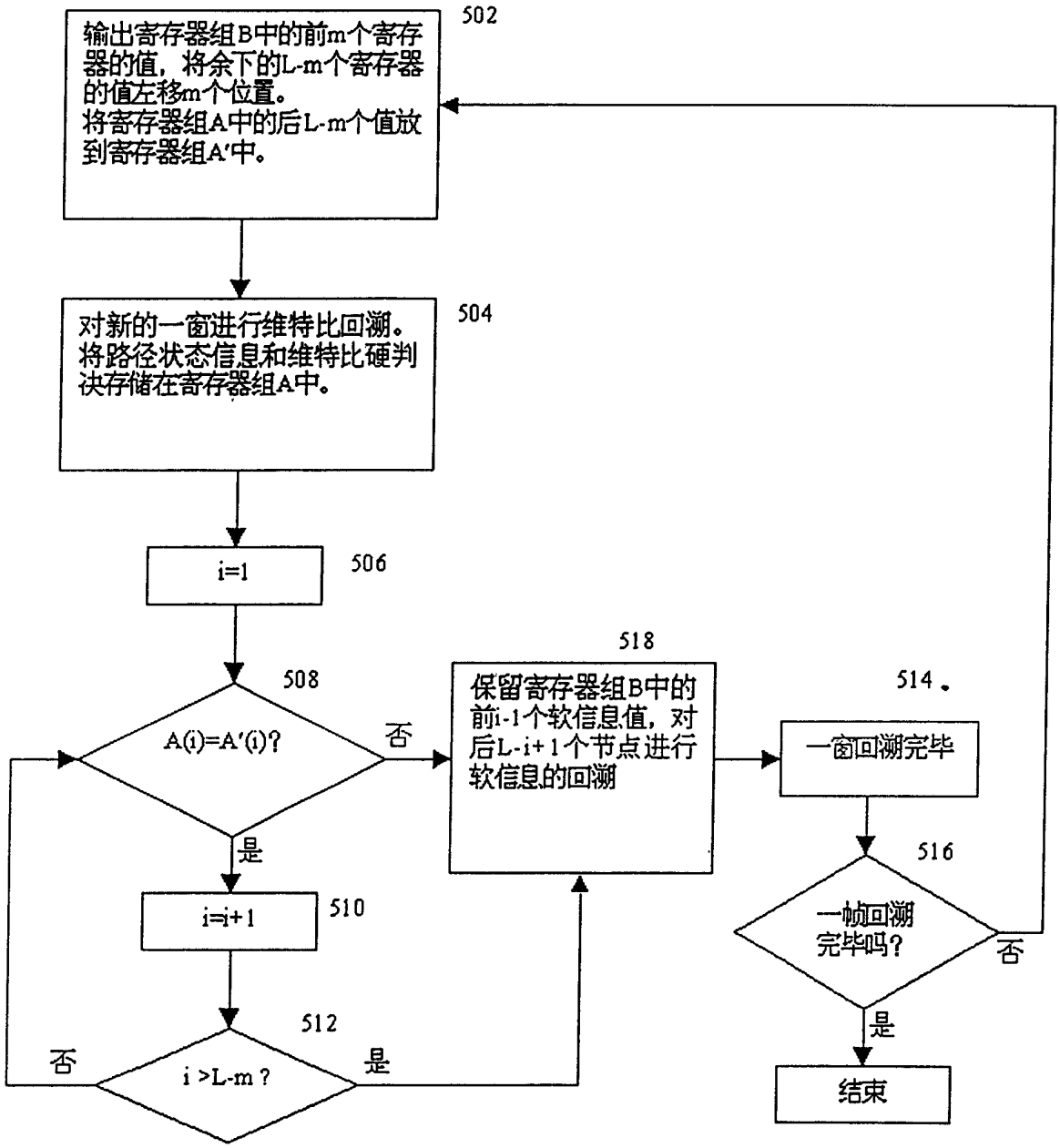

输出寄存器组B中的前m个寄存器的值，将余下的L-m个寄存器的值左移m个位置。
将寄存器组A中的后L-m个值放到寄存器组A'中。
502
对新的一窗进行维特比回溯。
将路径状态信息和维特比硬判决存储在寄存器组A中。
504
i=1
506
A(i)=A'(i)?
508
否
是
保留寄存器组B中的前i-1个软信息值，对后L-i+1个节点进行软信息的回溯
518
一窗回溯完毕
514
一帧回溯完毕吗?
516
否
是
结束
i=i+1
510
i >L-m ?
512
否
是

图 5

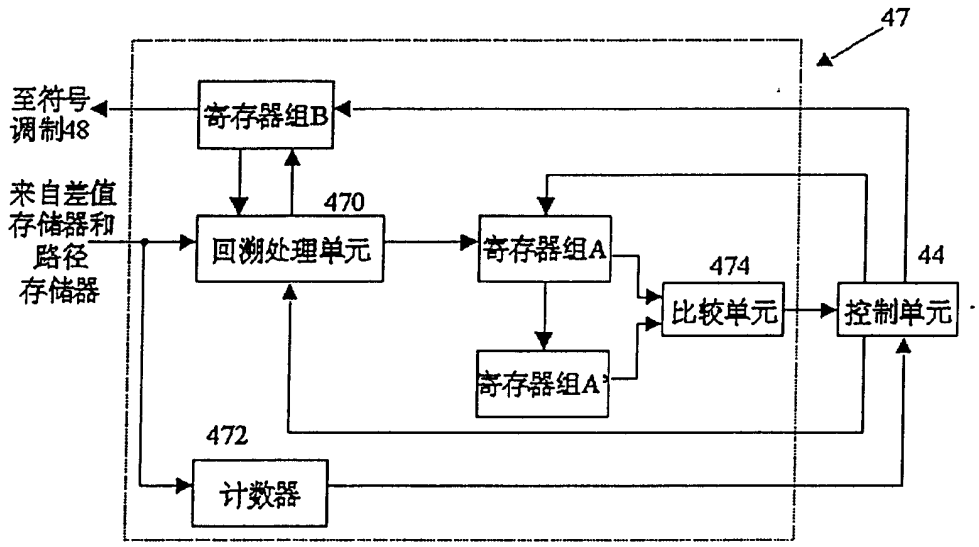

47
至符号调制48
寄存器组B
来自差值存储器和路径存储器
470
回溯处理单元
寄存器组A
474
比较单元
44
控制单元
寄存器组A'
472
计数器

图 7