

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>

H04N 7/26

H04N 7/30

G06T 9/40



# [12] 发明专利申请公开说明书

[21] 申请号 200410000130.1

[43] 公开日 2005 年 7 月 20 日

[11] 公开号 CN 1642278A

[22] 申请日 2004.1.6

[21] 申请号 200410000130.1

[71] 申请人 北京大学

地址 100871 北京市海淀区颐和园路 5 号

[72] 发明人 许 超

[74] 专利代理机构 北京君尚知识产权代理事务所

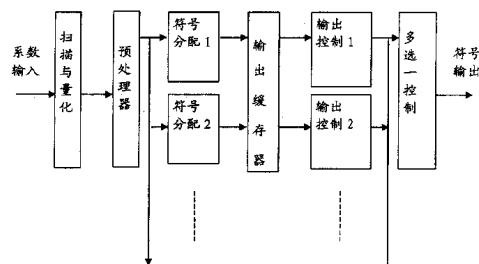
代理人 余长江

权利要求书 2 页 说明书 12 页 附图 4 页

[54] 发明名称 嵌入式零树编码方法及其电路结构

[57] 摘要

本发明涉及一种嵌入式零树编码方法及其电路结构。将嵌入式零树编码分为两级处理，同时进行多个位平面并行的编码。在第一级处理中，首先按照由低层子带向高层子带，逐组、逐层、逐树的扫描方式读取小波变换系数；然后将对应多个阈值的多级量化同时执行；量化后的数据进行预处理；预处理后的数据在各自的位平面中独立编码；数据编码产生的符号暂存于输出缓存器中；当一棵小波树的所有小波变换系数被分配了符号，并存入输出缓存器后，开始第二级处理；按照由高层子带向低层子带的扫描顺序，筛选有效的编码符号输出。可提高编码速度，减少编码时间，对集成电路的性能、资源要求较低，应用灵活且易于实现。可应用于视频数据压缩或图像数据压缩领域。



- 1、一种嵌入式零树编码方法，其步骤包括：
  - 1) 按照由低层子带向高层子带扫描顺序，逐组、逐层、逐树读入小波系数；
  - 2) 读入的小波变换系数同时执行对应多个阈值的多级量化；
  - 3) 多级量化产生的多个幅度位进行预处理，去除位平面之间的关联；
  - 4) 各位平面独立、并行编码，为所有结点分配符号；
  - 5) 待一棵小波树的结点全部分配符号后，由高层子带向低层子带扫描，筛选有效符号输出。
- 2、如权利要求 1 所述的嵌入式零树编码方法，其特征在于所述预处理为如小波变换系数的一个高幅度位是 1，就将其所有低幅度位清零，并做下标记。
- 3、如权利要求 1 所述的嵌入式零树编码方法，其特征在于所述编码为根据各个幅度位及其后代的性质分配相应的编码符号：当幅度位是 0 值，而它的后代不全是 0 值时，分配孤独零符号；当幅度位是 0 值，而它的后代也全是 0 值时，分配零树根符号；当幅度位是 1 值，而所属系数是正数时，分配正大系数符号；当幅度位是 1 值，而所属系数是负数时，分配负大系数符号。
- 4、如权利要求 1 或 3 所述的嵌入式零树编码方法，其特征在于所述分配的符号暂时存入输出缓存器中。
- 5、如权利要求 1 所述的嵌入式零树编码方法，其特征在于所述有效符号指该符号不属于有效零树根符号的后代，而且不属于高位平面中的大系数。
- 6、如权利要求 1 所述的嵌入式零树编码方法，其特征在于交替或顺序输出各位平面的有效符号。
- 7、一种嵌入式零树编码电路结构，包括扫描与量化模块、预处理器模块、一个或多个符号分配模块、一个或多个输出控制模块，其特征在于扫描与量化模块扫描、读取小波变换系数，进行多级量化；预处理器模块用于去除位平面之间的关联；符号分配模块对各位平面独立、并行编码，为所有结

点分配符号；输出控制模块筛选有效符号输出。

8、如权利要求 7 所述的嵌入式零树编码电路结构，其特征在于所述预处理器模块将一个高幅度位是 1 的小波变换系数的所有低幅度位清零，并做下标记。

9、如权利要求 8 所述的嵌入式零树编码电路结构，其特征在于所述预处理器包括“比特非与”和“比特或”逻辑电路，“比特非与”逻辑电路使小波变换系数只保留量化后的最高的幅度位 1，其低幅度位全部被清为 0，处理后的幅度位用来并行编码；“比特或”逻辑电路保留小波变换系数量化后的最高的幅度位 1，并将其低幅度位全部置为 1，处理后的幅度位用来作为低 1 幅度位的大系数标志。

10、如权利要求 7 所述的嵌入式零树编码电路结构，其特征在于各符号分配模块根据各个幅度位及其后代的性质分配相应的编码符号：当幅度位是 0 值，而它的后代不全是 0 值时，分配孤独零符号；当幅度位是 0 值，而它的后代也全是 0 值时，分配零树根符号；当幅度位是 1 值，而所属系数是正数时，分配正大系数符号；当幅度位是 1 值，而所属系数是负数时，分配负大系数符号。

11、如权利要求 7 或 10 所述的嵌入式零树编码电路结构，其特征在于还包括缓存器，暂时存入符号分配模块产生的小波变换系数编码；然后输送到输出控制模块。

12、如权利要求 7 所述的嵌入式零树编码电路结构，其特征在于还包括多选一控制模块，控制各位平面有效符号的交替或顺序输出。

## 嵌入式零树编码方法及其电路结构

### 所属技术领域:

本发明属于视频数据压缩或图像数据压缩领域,尤其涉及一种嵌入式零树编码方法,及其电路结构。

### 背景技术:

在视频数据压缩和图像数据压缩中,为提高压缩性能经常使用数学变换技术。目前经常使用的一种数学变换是小波变换,图1、图2为小波变换频带示意图,其中,L表示低频,H表示高频。图1是图像的1层小波变换,它产生4个子带。图2是图像的金字塔形4层小波变换,是在图像压缩领域常用的一种小波变换。它是在1层小波变换之后,对低频子带LL1再次进行第二层的小波变换,然后再对产生的LL2进行第三层的小波变换,最后再对产生的LL3进行第四层的小波变换。

图像经小波变换后,其小波变换系数(简称小波系数)有两个特点:第一、能量主要集中在低频子带;其余区域的能量较少,而且主要集中在与图像灰度突变边缘对应的小波系数处,且一般高层的系数比相应的低层系数的绝对值大。第二、同一方向上不同层的子带之间存在自相似性。这种层间的自相似,一方面,反映在对应于边缘的各层大系数位置间的相似;另一方面,也反映在小系数的层间关系上,与一个高层的小系数相应的低层系数一般更小。

为了充分利用小波系数的这两个特点,J. M. Shapiro提出了小波系数的嵌入式零树编码方法(见1.“Embedded image coding using zerotrees of wavelet coefficients”,1993, IEEE Trans. on Signal Processing. 1993, 41(12): 3445~3462. 2. “A fast technique for identifying zerotrees in the EZW algorithm”. 1996, Proc. IEEE Conf. on Acoustics, Speech and Signal Processing, Atlanta, Georgia, USA, 1996, 1455~1458)。这是一种高效的编码方法,而且码流具有嵌入式的性质。即:可以随时截断,精确地达到所需的压缩比,而解码、恢复出的图像可以保持较完整的全局信息。它的实现方法是,对

小波系数逐次量化，每次量化后，寻找同一方向上为零的小波系数，用固定的树状结构进行组织和编码，即所谓“零树编码”。

小波系数的树状结构，如图2中所示，用线连起来的小波系数构成小波树。其中位于高层子带的系数称为父结点，低层子带相应位置的系数叫做后代结点。位于最高层高频子带的一个结点和它的所有后代结点组成一棵小波树。一幅图像中，所有高频子带的小波系数沿各自方向形成一棵棵独立的小波树。

根据子带之间的自相似性，任一个为0的结点，它的后代结点也为0的概率较大。如果这种情况成立，就认为构成一棵零树，即，全是0值的树。利用这种零树编码，可以将0值小波系数有效的组织和表达，而且小波树结构固定，不需另加说明，因而获得了较高的压缩性能。

零树编码的另一重要方法，逐次量化逼近，是通过逐渐降低小波变换系数的量化阈值，逐步完成量化和编码。其中，每一次量化都将每个小波系数变成0或1，所有这些1比特数就形成一个位平面。每一次编码对一个位平面执行。

逐次量化、编码使小波系数的较重要的数据（较大的数值部分）先行输出。因此，即使任意停止码流输出，根据已有的数据，仍然可以产生出粗糙，但是完整的图像，即所谓“嵌入式”性能。

每次量化后的编码过程包括主控和从属两个过程，主控过程对位平面中的零树结构进行判别和编码；从属过程进行大系数细化编码。

在主控过程中，使用孤独零、零树根、正大系数、负大系数等4种符号来表示位平面中的各个结点的比特及其与前辈、后代结点之间的关系。其中，孤独零表示某结点的比特是0值，但它的后代结点不全是0值。零树根表示某结点的比特和它的所有后代结点的比特全是0值，而且它的前辈结点不是零树根。正大系数表示某结点的比特是+1值，而负大系数表示某结点的比特是-1值。当1棵零树产生，1个零树根符号就代表了此结点和它的所有后代结点的比特0值。因为小波树内形成零树的概率较大，所以使用零树根符号编码可以获得较高的压缩性能。

不过，根据零树根的定义，一个结点是否为零树根，不仅要判断其所有后代结点，还要判断其所有前辈结点，因而需要较复杂的递归处理。

零树根的数目直接关系到数据压缩的性能。零树根数目越多，数据压缩性能越高。因此，零树编码需要尽量增加零树根的数目。

阻碍零树形成的因素是大绝对值系数的存在。对于大绝对值系数，在后面的位平面编码中，仍然可能为大系数，从而可能多次阻碍零树根的产生。Shapiro 设计了一种“大系数跳过处理”技术。一旦小波系数在某一位平面的比特值是 1，即成为大系数，则在后面的低位平面的编码中将被作为 0 值处理，而且不再分配符号。

随着阈值的降低，大系数越来越多。跳过处理使它们不能影响后面的零树根的产生，因而保证了压缩性能。但是，大系数不能只由一个主控过程的符号来编码，否则图像的精确度将不能保证。因此，附加一个从属过程，将大系数的低幅度位依次输出。

从属过程的实现较为简单。而跳过处理的实现有一定难度。随着由高位平面向低位平面的编码，大系数越来越多，而且大系数的位置没有规则，要让电路不断地调整、跳过，难度较大。一般的解决方法是，一旦一个结点成为大系数之后，就将其所有的低幅度位清除为 0 值，并参加所有低位平面的编码。然后，增加一个后处理电路，对编码符号再进行筛选。

这种跳过处理也需要递归处理，例如，当一个结点是前大系数，需要跳过，但是清为 0 值并编码后，若编码成零树根，简单地跳过这个符号，它的后代结点将无法解码。因此，跳过之后，必须对其子结点补偿输出 4 个零树根符号；而如果某个子结点也是前大系数，那么，还需对相应的孙结点再补偿输出 4 个零树根符号，依此类推。仅当它被编码为孤独零，可以简单地跳过。

由上所述，嵌入式零树编码算法中的零树根符号编码和大系数跳过处理都要进行较繁复的递归处理，有待于通过简捷的方法来实现。

另外，嵌入式零树编码采用逐次量化、编码方法，而且每次量化后大系数需要清 0 及跳过处理，因此进行串行式的逐位平面编码较为容易。但是，量化的次数一般在 10 比特以上，要串行执行 10 个以上的位平面的零树编码，对于要求快速数据压缩的应用将不得不使用处理速度非常高的集成电路。低成本的集成电路难以实现。因此，零树编码需要快速的编码方法，来降低对集成电路

性能的要求。

#### 发明内容:

本发明的目的在于提供一种快速、简捷的嵌入式零树编码方法,使零树编码在视频数据压缩和图像数据压缩中可以快速实现,并降低对电路性能和资源的要求,应用灵活且易于实现。

本发明的另一目的在于提供一种实现上述方法的嵌入式零树编码电路结构。

本发明的嵌入式零树编码方法,其步骤包括:

1、按照由低层子带向高层子带扫描顺序,逐组、逐层、逐树读入小波系数;

2、读入的小波系数同时执行对应多个阈值的多级量化;

3、多级量化产生的多个幅度位进行预处理,去除位平面之间的关联;

4、各个位平面独立、并行编码,为所有结点分配符号;

5、待一棵小波树的结点全部分配符号后,由高层子带向低层子带扫描,筛选有效符号输出。

所述预处理为一旦发现小波变换系数的一个高幅度位是 1,就将其所有低幅度位清零,并做下标记,以保证预处理后的各个幅度位可以独立、并行地编码。

所述编码操作为根据各个幅度位及其后代的性质分配相应的编码符号;编码符号共有四种,孤独零、零树根、正大系数、负大系数;符号分配规则是,当幅度位是 0 值,但它的后代不全是 0 值时,分配孤独零符号;当幅度位是 0 值,而且它的后代也全是 0 值时,分配零树根符号;当幅度位是 1 值,而且所属系数是正数时,分配正大系数符号;当幅度位是 1 值,但是所属系数是负数时,分配负大系数符号。

所述分配的符号暂时存入输出缓存器中。

所述有效符号指该符号不属于有效零树根符号的后代,而且不属于高位平面中的大系数。

上述各位平面的有效符号是交替或顺序输出的。

本发明的嵌入式零树编码电路结构,包括扫描与量化模块、预处理器

模块、一个或多个符号分配模块、一个或多个输出控制模块，其中扫描与量化模块扫描、读取小波变换系数，进行多级量化；由预处理器模块对量化后的各幅度位进行清零和标记；符号分配模块对预处理后的各幅度位并行编码，分配相应符号；输出控制模块筛选有效符号输出。

本发明的嵌入式零树编码电路结构还包括缓存器，一棵小波树的所有小波系数编码数据暂时存入缓存器；然后输送到输出控制模块。

所述预处理器包括“比特非与”和“比特或”逻辑电路，“比特非与”逻辑电路使小波变换系数只保留量化后的最高的幅度位 1，其低幅度位全部被清为 0，处理后的幅度位用来并行编码；“比特或”逻辑电路保留小波变换系数量化后的最高的幅度位 1，并将其低幅度位全部置为 1，处理后的幅度位用来作为低 1 幅度位的大系数标志。

本发明的嵌入式零树编码电路结构还包括多选一控制模块，控制各位平面有效符号的交替或顺序输出。

本发明方法将嵌入式零树编码过程分为两级处理，第一级处理对一棵小波树的所有小波变换系数用 4 种符号编码；第二级处理筛选有效符号输出。

首先，按照由低层子带向高层子带扫描顺序、逐组、逐层、逐树读入小波系数。并将对应多个阈值的多次量化同时执行。量化后数据进行预处理，去除位平面之间的关联。然后各个位平面独立编码，以孤独零（00）、零树根（10）、正大系数（01）、负大系数（11）四种符号为所有结点编码。将编码符号输入缓存器中暂存。待一棵小波树的结点全部分配符号后，再按照由高层子带向低层子带的扫描顺序，筛选有效的编码符号输出。

每个位平面的编码使用四种编码符号，孤独零（00）、零树根（10）、正大系数（01）、负大系数（11）。其低比特对应小波系数量化后数值，当为 0 时，表示量化后为小系数；当为 1 时，表示量化后为大系数。其高比特略为复杂，当为 0 时，表示要么不是零树，要么是正大系数；当为 1 时，表示要么是零树，要么是负大系数。

根据前面算法介绍中符号的定义，正大系数（01）、负大系数（11）的分配完全由系数自身数值决定；孤独零（00）则由其自身和后代结点共



同决定；而零树根（10）则不仅依赖自身和后代结点的性质，同时还受前辈结点的影响。

对于大系数的跳过处理，由于前大系数被作为 0 值编码，分配的符号只能是孤独零或零树根。为孤独零时，可以直接跳过；为零树根时，必须在其子结点位置补偿分配 4 个零树根；若某个子结点也是前大系数，则要在相应孙结点位置再补偿分配 4 个零树根，依此类推。可见，无论是编码，还是跳过处理，递归处理总是针对零树根符号。

采用两级处理方法可以避免零树根符号所需要的递归处理。具体的方法是，在第一级处理中，由低层子带向高层子带逐组、逐层、逐树进行扫描，同时给所有结点分配符号。所谓“逐组、逐层、逐树”中的组、层和树是指，每个结点的 4 个子结点组成一组；每层与小波变换的层次相一致，但只限于一棵小波树内；每棵树由最高层高频子带中的一个结点和它的所有后代结点组成。第一级处理为所有结点分配符号，第二级处理就不再需要符号分配。

第一级处理中，由于遵循从后代结点向前辈结点依次、单向、规则的扫描编码，孤独零、正大系数、负大系数符号都可以确定。但是零树根不能完全确定，因为单向编码仅执行了零树根符号定义的一部分。即，当此结点的比特和它的所有后代结点的比特全是小系数 0，分配零树根符号，而没有考虑其前辈结点的性质。而且是给所有结点分配符号，必然产生冗余的零树根符号和应跳过的分配给前大系数的符号。

跳过冗余的符号，筛选有效的符号输出，由第二级处理完成。第二级处理采用由高层子带向低层子带扫描顺序，这是根据零树解码的要求，先输出前辈结点的符号，后输出后代结点的符号，以顺序地恢复小波系数。

跳过冗余的符号，也是根据前辈和自身结点的性质，跳过冗余的零树根符号和分配给前大系数的符号，也需要从高层向低层的扫描。其顺序与输出过程的一致，因此，可以将其嵌入到输出过程中，减少额外的扫描。

具体说来，在第二级处理中，当从高层向低层扫描时，若发现 1 个零树根符号，如果它不是前大系数，那么是有效的编码符号，予以输出。而它的所有后代结点符号都不再需要输出。若它是前大系数，那么是无效的

符号，它的 4 个子结点的符号必须输出。但是，若某个子结点也是前大系数，则同其父结点的处理，放弃无效符号，输出相应 4 个孙结点的符号。依此类推。对于孤独零符号，若它是前大系数，只要简单地跳过。

归纳起来，第一级处理是根据结点自身数值及其后代结点数值，分配编码符号。由于第一级处理为所有结点分配了符号，第二级处理只需筛选有效的符号输出。这两级处理都不需要递归处理，只需要单向处理，因此，它是一种简捷、有效的零树编码方法。

简捷、快速地实现单个位平面的零树编码，仍不足以满足以低成本集成电路快速编码的要求。由于 10 个以上的位平面存在，逐个位平面的串行编码将消耗大量的运算时间。

并行处理是一种常用的减少运算时间的方法。但如何进行并行处理，对运算时间、成本影响很大。前面提到，小波树之间是相互独立的，因此可以直接进行并行处理。但是，从一个存储器中同时读写多个小波树的数据，将给固定带宽的数据传输带来较大的困难。

本发明提出多个位平面并行处理，使用同一棵小波树的数据，不改变数据传输带宽。虽然，由于大系数的跳过处理，高位平面的零树编码影响低位平面的零树编码，要实现多个位平面的并行处理，需要设法去除位平面之间的关联，但是，通过一个巧妙而简单的预处理器，对前大系数的低幅度位进行清零，并进行标记，就可以保证各个位平面独立、并行地进行编码。

具体的预处理器的操作包括两部分，一是“比特非与”逻辑运算，将每个高幅度位取“反”，再和每个低幅度位相“与”。当任一高幅度位是 1 时，“比特非与”操作将比它低的所有幅度位清成 0 值。二是“比特或”逻辑运算，将每个高幅度位和每个低幅度位相“或”。当任一高幅度位是 1 时，“比特或”操作将比它低的所有幅度位置成 1 值。因此，“比特非与”操作后的每个幅度位可以独立地在各自的位平面中编码，而“比特或”操作后每个幅度位可以作为其低一幅度位的前大系数标记。标记为 1 时，此幅度位 0 值属于前大系数，应该跳过。否则，正常编码。

根据上述零树编码方法，本发明设计了相应的电路结构，如图 5 所示。

扫描与量化模块控制小波系数由外部存储器的输入，使按照由低层子带向高层子带逐组、逐层、逐树的扫描顺序。小波系数进入零树编码电路后，先执行多级量化。然后由预处理器对量化系数进行预处理，使各幅度位可以独立编码。符号分配模块给每个预处理后的幅度位分配 1 个符号，仅根据其自身及后代结点数值。各个符号依次存入输出缓存器中。待一棵小波树的所有系数都已经分配了符号，并进入了输出缓存器后，输出控制模块开始工作。按照由高层子带向低层子带的扫描顺序，输出控制模块筛选各自位平面的有效符号输出。多选一控制模块使各个位平面的输出交替进行。

如果有必要简化电路结构，输出控制模块的数目可以减少。例如，使用 1 个输出控制模块，见图 6 所示，将多选一控制模块提到输出控制模块之前，顺序选择多个位平面的符号输出给输出控制模块，再筛选有效符号输出。

本发明的关键在于：一是两级处理，可以去除嵌入式零树编码中的递归处理，使编码过程简捷。二是多个位平面的并行编码，可以大幅度地提高编码速度。使用本发明技术，可以快速实现嵌入式零树编码，以及由低成本的集成电路实现快速编码。

本发明所提供的零树编码方法，具有以下的特点：

1) 该零树编码方法充分利用零树编码的特点，设计了从低层到高层，逐组、逐层、逐树的扫描、编码，以及从高层到低层，从前辈结点到后代结点的扫描、筛选和输出。去除了较繁复的递归处理。而且并行处理多个位平面的零树编码。既保证了零树编码算法的正确实现，同时大大降低了运算复杂度和运算时间，可以较容易地实现快速的视频、图像压缩编码，同时保证对集成电路的较低要求。

2) 该零树编码采用以小波树为单位，而不是对整幅图像或多个小波树进行零树编码，可以大大减少对内部存储器的容量要求，以及数据传输带宽的要求。

3) 从电路设计角度而言，采用两级处理，去除递归处理，可以大大减少对存储器的交互读写，而交互读写将增加控制电路的复杂度和资源消耗。

4) 成本低廉。它对集成电路性能和资源的要求相对很低，并且不需要

专门的电路结构和大容量片内存储器，因而能够集成到一片低成本的集成电路之中。

5) 可靠性好。由于结构简单、紧凑，具有良好的实用性和可靠性。

6) 通用性强。可完成任意常用尺寸的图像的快速零树编码。

本发明可以同时、并行编码多达 16 个以上的位平面，根据本发明方法设计的零树编码电路将是一个对电路性能、资源要求较低的、可实时完成任意常用尺寸图像零树编码的快速、低成本的集成电路。

附图说明：

图 1 为二维小波变换频带示意图。

图 2 为四层小波变换频带分布示意图。

图 3 为本发明符号分配示意图。

图 4 为本发明零树编码电路扫描图。

图 5 为本发明零树编码的电路框图。

图 6 为本发明零树编码的简化电路框图。

图 7 为本发明零树编码的第一级处理的部分电路示意图。

具体实施方式：

下面结合附图及具体实施例对本发明再作进一步的说明。

本发明是一种实现嵌入式零树编码的快速方法和电路结构。它的工作流程是，首先从低层到高层，按照逐组、逐层、逐树的扫描顺序，将数据读入零树编码电路；进行多级量化和预处理；然后，对预处理后的各个位平面并行编码；编码符号暂时存入输出缓存器中；待一棵小波树的系数全部分配符号后，再按照由高层子带向低层子带，对于输出缓存器，为后入先出的扫描顺序，筛选有效的编码符号输出。

小波树的输入扫描、编码方式如图 4 右部所示。如果不采用本发明的两级处理，常用的扫描、编码方式如图 4 左部所示。显然，本发明采用的扫描、编码方式更为规则和简捷。

以图 2 所示的四层小波变换为例，具体说明本发明的电路结构的工作原理及本发明的零树编码快速方法的实现。

本发明的零树编码电路结构如图 5 所示。它从第一层的后代结点开始扫描，将四个属于同一父结点的一组子结点顺序读取。一组读完后，读取下一组的结点。待第一层的 16 组读完后，读取第二层相应的 4 组。然后，读取第三层相应的 1 组 4 个结点。最后，读取第四层唯一的 1 个结点。

每个结点被读入零树编码电路后，马上进行多级量化。量化后，立刻输入预处理器进行预处理。

预处理器的电路结构如图 7 左部所示，它主要执行两个操作。一是“比特非与”操作，只要小波系数在某一高位平面出现一个幅度位 1，即成为大系数，则将所有低幅度位清为 0，否则保持不变。操作后的各位平面的幅度位称之为大系数清 0 值，彼此之间独立，直接用来并行编码。二是“比特或”操作，只要小波系数在某一高位平面出现一个幅度值 1，则将所有低幅度位置为 1，否则保持不变。操作后的各位平面的幅度位作为其低一幅度位的前大系数标记。

预处理后的分属不同位平面的数据送入各自的符号分配模块进行编码。单个符号分配模块的电路结构如图 7 右部所示。

本发明的两级处理中，符号分配规则如图 3 所示。首先判断预处理后的幅度位是否为 1，如果是，则看该系数是否为正数？如果幅度位为 1 且为正数，则输出正大系数符号；如果幅度位为 1 但为负数，则输出负大系数符号；如果幅度位为 0，则看其后代的幅度位是否都为 0，如果是，则输出零树根符号；如果幅度位为 0 但其后代的幅度位不全为 0，则输出孤独零符号。

零树编码电路中，后代的幅度位的判别是将顺序输入的一组 4 个结点的大系数清 0 值通过 D 触发器同步后，进行“比特或非”操作，其输出值再与 ZTR 标识寄存器中对应的此 4 个结点的高比特值进行“比特与”操作，结果为 1，表示为零树，否则不为零树。产生的 1 或 0 再输出到 ZTR 标识寄存器。对于最低层的结点，没有后代结点，可以假设具有全 0 的后代结点。

结点符号的分配，首先根据自身幅度位，其为 1，选择其正、负号作为符号的高比特值；其为 0，选择 ZTR 标识寄存器中对应值作为符号的高比

特值。因此，将自身比特同其正、负号做“比特与”运算，自身比特取反后同 ZTR 标识寄存器中对应值做“比特与”运算，两个结果做或运算就产生符号的高比特值。符号的低比特值就等于自身比特值。具体的电路结构如图 7 右上角所示。

产生的符号与大系数标志一起写入输出缓存器。待一棵小波树的所有编码符号写入输出缓存器后，输出控制模块开始工作。按照从高层到低层的顺序进行符号扫描，即后入先出的原则，选择有效符号输出。

输出控制的规则依据零树根符号定义和跳过处理的双重要求而确定。当且仅当其祖先不是有效的零树根符号，而且其大系数标志是 0 时，编码符号有效，予以输出。即，如果其祖先是有效的零树根符号，或者如果其大系数标志是 1 时，都属于无效符号，不能输出。

在要求编码速度较高的零树编码应用中，使用多个输出控制模块并行工作，如图 5 所示。按照高位平面的符号优先传输的原则解决总线竞争，通过多选一控制模块，将符号分别、依次输出。

对于要求编码速度略低的零树编码应用中，可以只使用一个输出控制模块，如图 6 所示。多选一控制模块提到输出控制模块之前，按照从高位平面到低位平面的原则，依次输出各个位平面的编码符号。

根据上述，该零树编码过程主要包括以下步骤：

1) 小波系数输入。按照从低层到高层，逐组、逐层、逐树的扫描顺序，将小波系数读入零树编码电路。

2) 量化。根据码率控制的要求定出多个量化阈值，对小波系数进行同时、多级量化。

3) 预处理。如果小波系数在某一高位平面出现一个幅度位 1，即认为其在此位平面成为大系数，对于其所有的低幅度位清 0。同时，对大系数进行标记。

3) 符号分配。两级处理中的符号分配规则是，(1) 当某结点的大系数清 0 值为 +1 或 -1 时，分配正大系数或负大系数符号。(2) 当某结点和它后代结点的大系数清 0 值全是 0 时，分配零树根符号。(3) 当某结点的大系数清 0 值是 0，而它后代结点的不全是 0 时，分配孤独零符号。所有符号

存入输出缓存器。

4) 符号输出。按照从高层到低层的顺序进行符号扫描, 若其祖先不是有效的零树根符号, 而且大系数标志是 0 时, 输出此编码符号。

以上所述, 仅为本发明的较佳实施例而已, 并非用于限定本发明的保护范围。

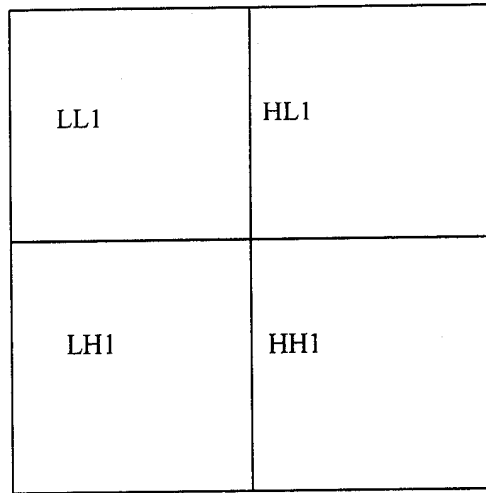


图 1

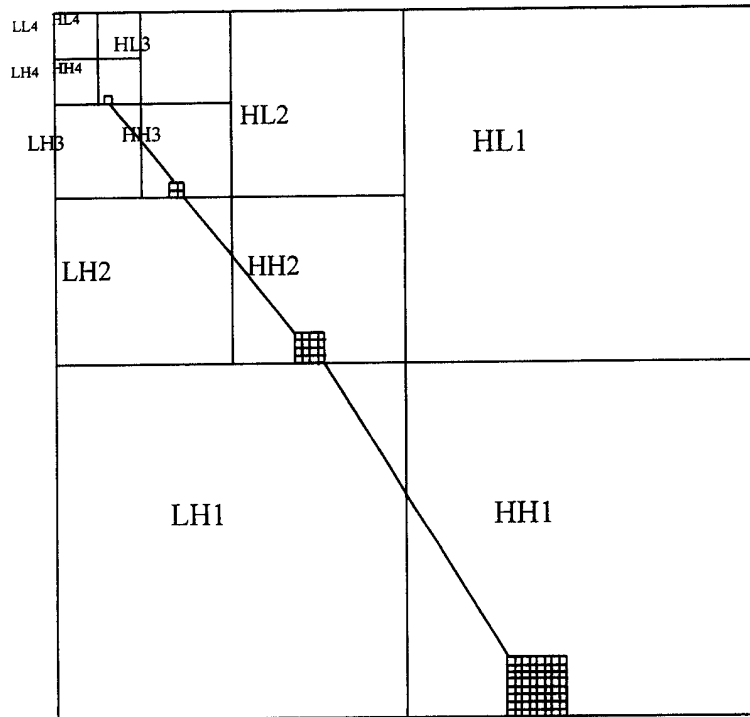


图 2



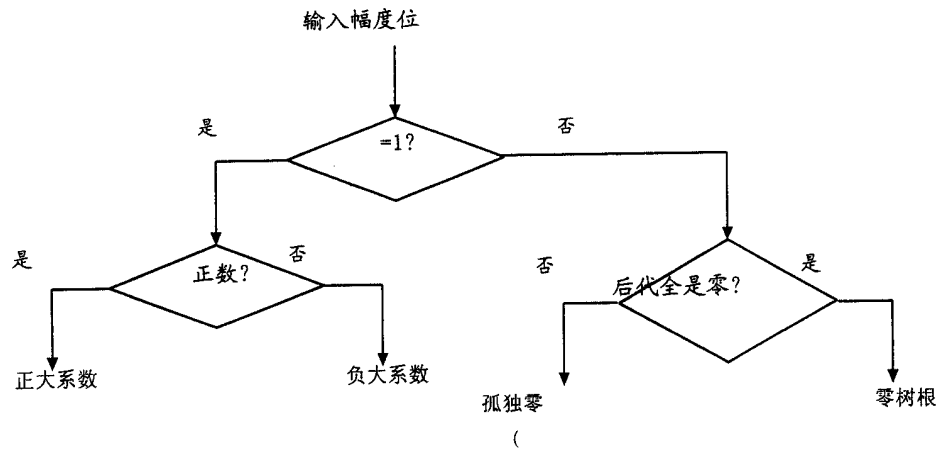


图 3

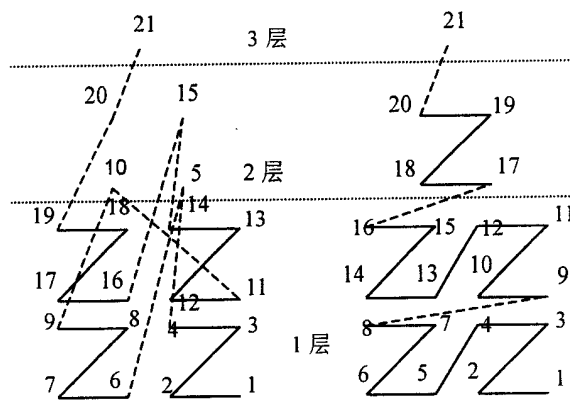


图 4

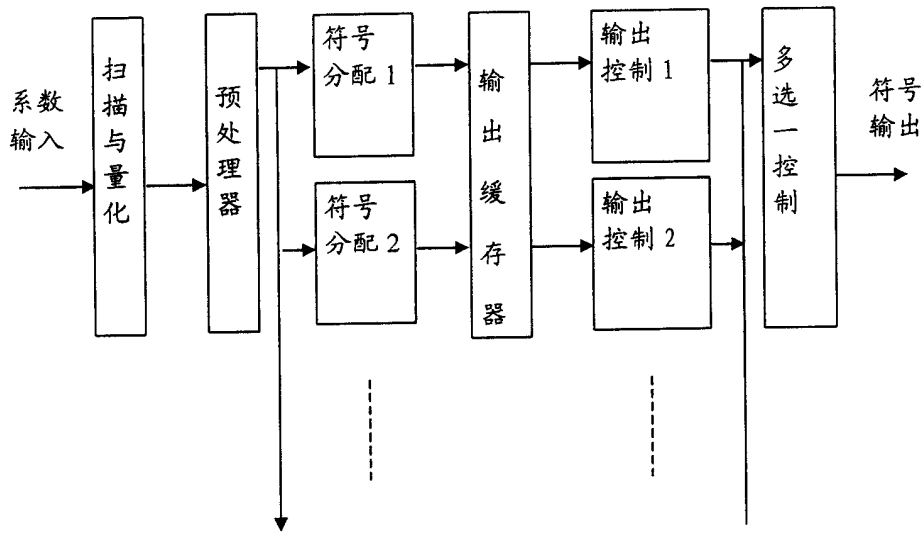


图 5

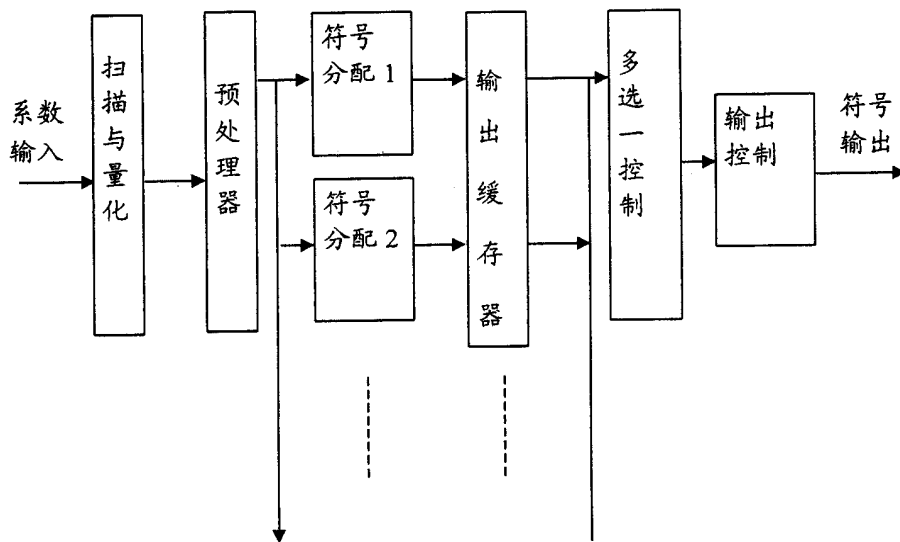


图 6

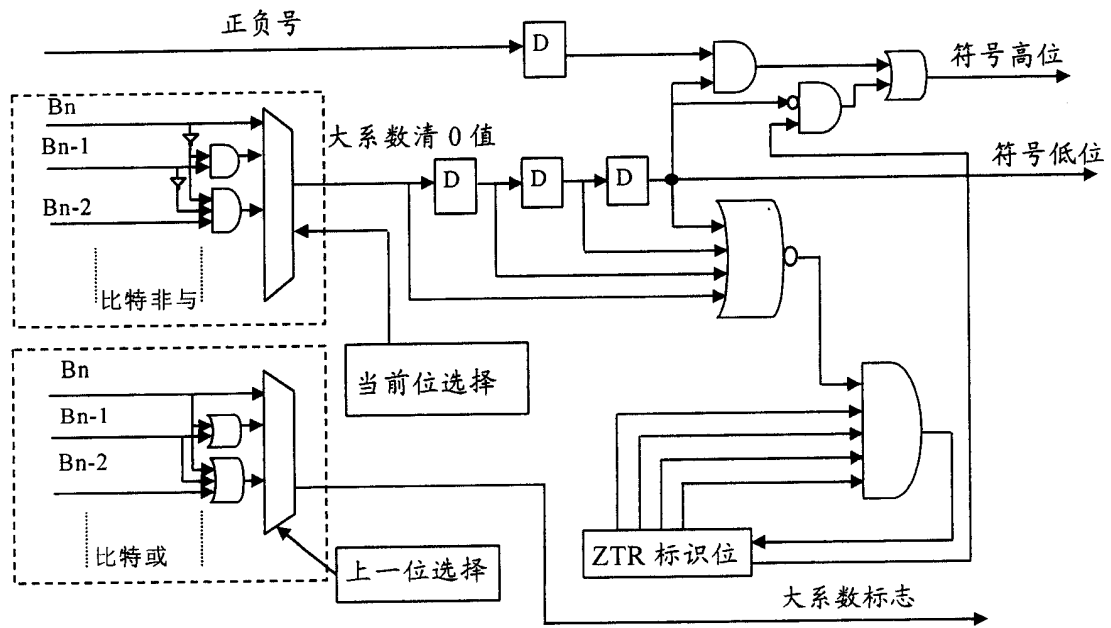


图 7