[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04N 7/26
H04N 5/272

[12] 发明专利说明书

[21] ZL 专利号 99119091.2

[45] 授权公告日 2004 年 5 月 26 日

[11] 授权公告号 CN 1151679C

[22] 申请日 1999.9.15 [21] 申请号 99119091.2
[30] 优先权
[32] 1998. 9.17 [33] FR [31] 9811627
[71] 专利权人 汤姆森多媒体公司
地址 法国布洛里
[72] 发明人 埃利亚斯·布苏莱蒙
审查员 郑 直

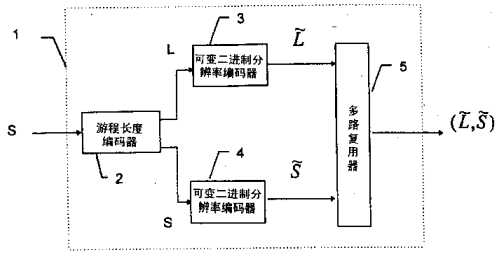
[74] 专利代理机构 中科专利商标代理有限责任公司
代理人 朱进桂

权利要求书 3 页 说明书 15 页 附图 8 页

[54] 发明名称 图形数据压缩方法和装置以及数字视频译码器

[57] 摘要

图形数据压缩的方法，被用于以点模式(位图数据)对图像编码，数据由被称为符号 S 的变换表中的一系列地址组成，其特征在于：它包括下面的步骤：－对符号进行游程长度编码(2)，以便就每一符号顺次相同的范围给出与相同符号的数相应的数 L，数 L 与符号 S 相关联；对 L 进行可变二进制分辨率编码(3)，作为 L 值的函数。 各种应用特别涉及 OSD(在屏显示)数据的编码和译码。



ISSN1008-4274

1. 用于以点模式对图像编码的图形数据压缩方法，数据包括被称为符号 S 的颜色查寻表的一系列地址，其特征在于：它包括下面的步骤：

将不相同的符号存储在字典（16）中；

根据符号在字典（16）中的地址，对符号编码；

对字典中的地址进行可变二进制分辨率编码（13），作为符号编码。

2. 根据权利要求 1 的方法，其特征在于：它包括下面的步骤：

对符号进行游程长度编码（2），给出符号 S 的游程长度 L；

对 L 进行可变二进制分辨率编码（3），作为 L 值的函数。

3. 根据权利要求 1 的方法，其特征在于：图形数据是 OSD 数据，其中图像是调色板化的图像。

4. 根据权利要求 2 的方法，其特征在于：位图数据的压缩产生编码数据流数据，包括：

可变范围代码字段，它由确定游程长度字段范围的第一字段，和确定符号字段范围的第二字段组成；

用于编码 L 的游程长度字段，它的范围与用于 L 编码的二进制分辨率相对应；

用于编码 S 的符号字段，它的范围与对 S 编码的二进制分辨率相对应。

5. 根据权利要求 4 的方法，其特征在于：

用于编码 L 的二进制分辨率，由在 L 编码的各位之前所派送的多个零确定，零的数目是对 L 值编码所用的位数的函数，

根据其在字典中的地址对符号编码所用的二进制分辨的改变，由在地址编码的各位之前所派送的预定的多个零的数目确定（27），其数目大于用于编码 L 的相继的零的最大数目。

6. 用于以点模式对图像编码的图形数据压缩装置，数据由在数据流中派送的被称为符号 S 的变换表中的一系列地址组成，其特征在于：它包括游程长度编码器（2），该编码器（2）接收数据流，以便就每个游程长

度提供与符号 S 相关联的数 L；分级可变二进制分辨率编码器（3），用于对 L 编码；可变二进制符号编码器（4），包括以存储各种符号的字典（15）；以及在字典中地址的可变二进制分辨率编码器（13），用于对 S 编码。

7. 用于对根据权利要求 2 中所述的方法压缩的图形解压缩的装置，其特征在于：它包括存储器（35），用于记忆存储字典；接收压缩数据流的字段识别电路（34），用于检测 L 和 S 的二进制分辨率，并提取 L 和 S 的编码值；符号 S 提取电路（36），该提取电路被链接至存储器（35），用于按地址派送 S 的编码值并回取地址中的内容；游程长度译码电路（37），用于对 L 的编码值译码；生成符号序列的电路（38），它接收与符号 S 有关的已译码的游程长度 L 的值和已译码的符号 S 的值，以便在数据流中发送 L 次的 S 值。

8. 数字视频译码器，它用于接收包括图形数据的数据流，其特征在于：它包括用于以点模式对图像编码的图形数据压缩装置（1），数据由在数据流中派送的被称为符号 S 的变换表中的一系列地址组成，图形数据压缩装置（1）包括游程长度编码器（2），该编码器（2）接收数据流，以便就每个游程长度提供与符号 S 相关联的数 L；分级可变二进制分辨率编码器（3），用于对 L 编码；可变二进制符号编码器（4），包括以存储各种符号的字典（15）；以及在字典中地址的可变二进制分辨率编码器（13），用于对 S 编码；

用以存储被压缩的图形数据的存储器；

图形数据解压缩装置，用于解压缩使用以点模式对图像编码的编码方法压缩的图形数据，所述图形数据包括被称为符号 S 的颜色查寻表的一系列地址，所述方法包括下面的步骤：

将不相同的符号存储在字典（16）中；

根据符号在字典（15）中的地址，对符号编码；

对字典中的地址进行可变二进制分辨率编码（13），作为符号编码；

对符号进行游程长度编码（2），给出符号 S 的游程长度 L；

对 L 进行可变二进制分辨率编码（3），作为 L 值的函数；

用以解压缩将被传送到显示电路的数据。

9. 数字视频译码器，它用于接收包括使用以点模式对图像编码的编

码方法压缩的图形数据的数据流，所述数据包括被称为符号 S 的颜色查寻表的一系列地址，所述方法包括下面的步骤：

将不相同的符号存储在字典（15）中；

根据符号在字典（15）中的地址，对符号编码；

对字典中的地址进行可变二进制分辨率编码（13），作为符号编码；

其特征在于数字视频译码器包括：用以存储这样的图形数据的存储器；图形数据解压缩装置，用于解压缩使用以点模式对图像编码的编码方法压缩的图形数据，所述图形数据包括被称为符号 S 的颜色查寻表的一系列地址，所述方法包括下面的步骤：

将不相同的符号存储在字典（15）中；

根据符号在字典（15）中的地址，对符号编码；

对字典中的地址进行可变二进制分辨率编码（13），作为符号编码；

对符号进行游程长度编码（2），给出符号 S 的游程长度 L；以及

对 L 进行可变二进制分辨率编码（3），作为 L 值的函数；

所述图形数据解压缩装置接收存储的数据，以解压缩将被发送到显示电路的数据。

图形数据压缩方法和装置以及数字视频译码器

技术领域

本发明涉及图形数据的实时编码和译码方法和实时编码／译码器（codec）。特别是它应用于屏幕显示数据的编码／译码，较好的是已知的在屏显示或 OSD。这种显示型式被应用在视频译码器，电视接收机，录像机，和所有类型的视频装置中。

背景技术

OSD 显示可用于显示电视台的台标，供设备使用的菜单式目录以及程序所用的电子引导目录（程序引导），也可用于更复杂的应用，例如气象图背景的可视显示。它可以只是这些数据在屏幕上的可视显示，或者是与可视图像重叠在一起。

涉及这种 OSD 显示的数据，可以是驻留数据，那就是说，数据被永久地保存在使用这种显示型式的视频装置的存储器中，或者它可以采取硬件制造时被装入的信息形式，或者甚至是远程装入的信息。这些数据也可以是以一定距离被发送的视频数据流。这包括例如数字译码器。在 MPEG（活动图像专家组的首字母缩略词）型的视频数字译码器的情况下，即按照这个 MPEG 标准处理已编码的视频数据流（视频位流）的情况下，视频图像以压缩的形式被接收，特别是通过活动估算和离散余弦变换的虚似，利用视频序列的空间和时间冗余度。OSD 数据不涉及这种处理，它们是以未压缩形式被合并到 MPEG 数据流中，并以能实时使用的形式被存储在译码器中。

OSD 显示的应用变得越来越复杂，要求存储极大量的数据。由于费用的原因，分配给 OSD 数据的随机存取器 RAM 限制了这些用途，要不就会使所生成的图像或花型的清晰度降低，要不就限制了菜单式目录或各种应用

上的参差性。

发明内容

本发明的目的是补救上述这些缺点。

它的主题是用于以点模式（位图数据）对图像编码的图形数据压缩的方法，数据由被称为符号 S 的转换表中的一系列地址组成，其特征在于包括下述步骤:

－将各种不同的符号存储在字典中，

－根据其在字典中的地址，对符号进行编码，

－对字典中的地址进行可变二进制分辨率编码，用作符号编码。根据一个特定的装置，这种方法的特征在于包括下述步骤:

－对于相同的相继符号的每一范围，符号的游程长度编码给出一个与符号 S 相联系的数 L，它相应于相同符号的数目，

－对 L 进行可变二进制分辨率编码，作为数值 L 的函数。

本发明也涉及用于以点模式（位图数据）对图像编码的图形数据压缩装置，数据由数据流中输送的被称为符号 S 的转换表中的一系列地址组成，其特征在于它包括: 游程长度编码器，它依次接收数据流，对每个游程长度给出与符号 S 相联系的数 L; 可变二进制分辨编码器，分级对 L 编码; 可变二进制分辨率符号编码器，它包括用于记忆存储各种不同符号的字典和字典中的可变二进制分辨率编码器，用于对 S 编码。

本发明也涉及用于图形数据译码器的装置，其特征在于它包括: 存储器，用于存储字典; 字段识别电路，它接收被压缩的数据流，用于检测 L 和 S 的二进制分辨率，并提取 L 和 S 的编码值; 符号 S 提取电路，提取电路被链接至存储器，用于发送 S 的编码值作为地址，并回取地址的内容; 游程长度译码电路，用于对 L 的值进行译码; 用于生成符号序列的电路，它接收与符号 S 有关联的游程长度的译码值 L 和符号 S 的译码值，以便在数据流中发送 L 次的值 S。

被压缩的值一般涉及位图类型的数据。它们是有效像素值的地址，那就是说可视地在屏幕上显示的像素的亮度和／或色度值，它们被存储在被称为调色板的颜色调色板中。因此，图像被说成是经过调色的。

所以，压缩的两个基本步骤是:

－对相同符号的每一个序列的长度（被称为游程长度）进行可变分辨率编码。对长度 L 的相同的顺序符号的每一游程长度，每一个被检测到的重复序列的长度 L，被进行分辨编码，作为取决于游程长度 L 值的初始级的函数是可变的。分级编码使它有可能归于最佳（最小）码，它作为原始级 L 的函数是可变的。这些级别是根据图形的图像的大多数，用统计方法确定的。

－对符号本身进行可变二进制分辨率编码，作为它在适配字典中出现的次序的函数。实现这一点的方法是在它们被遭遇时，将各种不同的符号存储在字典中，并根据它们在字典中的实际地址对符号编码，这相应于随所碰到的颜色数目而增加的可变二进制分辨率编码。对于地址字来说，有效地址应理解为不予处理的不起作用的为零的较高位。

本发明的一个特征是在存储这种 OSD 信息之前，对它进行压缩，或者是在涉及驻留数据时，存储被压缩的信息。所用的压缩算法是简单的，可允许实时应用，通过用于 OSD 显示的图形编码的虚拟，而给出较高的压缩比。这一方面是因为在这种编码过程中所使用的颜色数目本身是有限的，以便减少要输送到录像机或要存储到这个译码器中的数据量。另外，OSD 显示一般表现为均匀的颜色区域和重复的图样。

因此，这种算法的完成是两种类型编码的组合：游程长度编码（RLC）和可变二进制分辨率编码，即对游程长度编码，和对符号也就是地址编码。通过本发明的虚拟，译码器，电视接收机，录像机等等所要求的随机存取存储器可以减少。对于给定的成本，硬件产品的成本可以降低，或者，涉及在屏显示的应用可具有更高的性能，和／或显示具有较好的质量。

附图说明

本发明的特征和优点，随着下面结合例子和参考附图的描述，将变得更为明显，附图中:

图 1 表示根据本发明的位图数据编码器，

图 2 表示对 L 编码的分级编码器，

图 3 表示符号 S 编码器，

图 4 表示位图数据编码方法的流程，

图 5 表示位图数据流的编码的例子，

图 6 表示位图编码数据的译码器，

图 7 表示位图编码数据的译码方法的流程，

图 8 表示位图编码数据流的译码的例子，

图 9 表示一个编码译码器。

具体实施方式

在目前存在许多对于 OSD 显示的清晰度，存储器和相应数据的使用的限定。一般来说，OSD 数据是以位图的形式被存储的，广泛应用表现为图像以点模式编码，也就是说，根据数据（亮度，色度）以像素为单位作图显示（图像）。调色板的存储器或寄存器包含生成符号和图形所需要的各种不同的颜色。通过分辨这些颜色而对这些颜色的矢量型寻址，不是通过它们的特征编码，而是通过它们在调色板寄存器中的地址来进行的。这个调色板寄存器也被称为颜色变换表或颜色查寻表（CLUT）。

OSD 信息一般被分配到图像上的部位或区域，区域由坐标确定。对于每个部位，对属于这个部位的像素进行编码所需要的所有颜色，与这个部位所用的调色板相对应。对于每个部位，位图给部位中的每个像素分配变换表中的地址，这个变换表包含这些像素的亮度和色度信息，即 Y、U、V。

下面叙述根据现有技术的数据存储的一个例子，所述数据是由 MPEG 译码器接收的，将要存储在这个译码器的随机存储器 RAM 中。RAM 存储器的第一部分存储被压缩的 MPEG 数据流，第二部分存储特定的 OSD 显示（OSD 数据），第三部分存储例如解压缩图像的视频数据，它们是需要的，例如按照 MPEG 标准给图像重新排序。

更详细地说，OSD 数据的存储器本身建构为三个区域，第一区域相应于坐标，第二区域相应于调色板，第三区域相应于位图数据。被分配给指定 OSD 部位的数据的存储器第一区域，用来存储坐标，便可以确定一个地方，在这里将显示这个 OSD 部位。第二区域用来存储表示这个 OSD 部位所需要的调色板，或者 CLUT 表，第三区域则用于每个像素，在变换表（CLUT）中的地址，相应于按照亮度和色度对像素的编码。在这个第三区域中的这

些数据被称为位图数据。

图 1 表示编码器 1，用于在存储于 RAM 存储器例如 MPEG 译码器的存储器之前，实现对 OSD 数据的编码和压缩方法。上述 OSD 数据事实上是位图数据。这些位图数据如上所述表现为一系列地址，它们将被称为符号 S，这些位图数据作为输入，被图中以破折号表示的并与电路 2 至 5 组合在一起的编码器 1 接收。这些数据被向前输送至固定二进制分辨率游程长度编码器 2。这个编码器在第一输出端提供 L 信息，在第二输出端提供 S 信息，这相应于对每个部位来说，所接收的输入是 L 个连续相同的符号 S。提供值 L 的输出被输送至分级可变二进制分辨率编码器 3，以便根据它的原始级对这个值 L 进行编码，而提供值 S 的输出被输送至可变二进制分辨率符号编码器 4。编码器 3 和 4 分别提供 L 和 S 的编码值$\tilde{L}$和$\tilde{S}$，并被链接至多路复用器 5 的输入端，多路复用器 5 的输出是编码器 1 的输出。

游程长度编码器 2 在编码方法的使用上有所延伸，比已知的游程长度编码或 RLC 好。游程长度编码的原理是计算顺次重复的第 K 个符号 S_K 的数 L_K，和以一对 $V_K = (L_K, S_K)$ 代替这个符号的重复序列。L_K 相应于同样的符号 S_K 的重复数目。然后，使用下述专门的编码方法单独地将该对的每个值编码。

值 L 的编码通过分级可变二进制分辨率编码器 3 进行，作为它的原始级的函数。这个编码器所采用的原理被表示在图 2 中。

来自游程长度编码器 2 的待要编码的信息 L 同时被输送至级别检测电路 6 和可变分辨率编码电路 7，后者提供从编码器 3 向前的 L 的编码值$\tilde{L}$。

在我们的例子中，级别检测电路由特定于每个级的 5 个比较器 8 组成，并且并行地接收被送到编码器 3 的输入端的信息，也就是说值 L（二进制编码）。每个比较器有 3 个输出，分别相应于比较器的输入的信息单元，即低于级限，高于级限，和处于特定于这个比较器的级别界限（或间隔）之中。比较器的输出被链接到逻辑电路 9，它将来自比较器的全部信息组合在一起。它有 5 个输出，它们也是级别检测电路 6 的输出，每级一个输出，被链接到可变二进制分辨率编码电路 7 的 5 个输出。这个编码电路 7 由代码转换器 10 组成，作为输入，它接收值 L（二进制编码）的相应信息的指示这个数值属于哪一级的 5 个输入。代码转换器的 5 个输出被链接到

多路复用器 11 的 5 个输出，多路复用器 11 的输出是编码器 3 的输出。这个输出提供相应于输入数据流 L 的编码数据流$\tilde{L}$。

在详述电路功能之前，我们将说明应用于分级编码的可变二进制分辨率编码的原理。

值 L 可能变化的范围是根据值 m 预先确定的，以使 $2^{2m+2}-1$ 相应于编码 L 的最大可能值（符号重复的最大次数）。这个范围下降为 m+1 个数值间隔，每个间隔由值 2^{2n} 和 $2^{2m+2}-1$ 界定，n 从 0 变化至 m，一个间隔相应于一级。分级可变二进制分辨率编码原理是使每一级有不同的分辨率属性。这个二进制分辨率相应于 L 编码所用的位（比特）数，因此，取决于值 L 所属的级别。这是一个在实际上要考虑的问题，在二进制字 L 中，只有“有效的”编码位，也就是不顾及为零的较高位，要知道，这些位不被传输就相当于它们的值为零。对于低的 L 值，编码是低的二进制分辨率（n=0，1），而对于较高的 L 值，编码值有较高的二进制分辨率（n=2，3，4）。

通过一个特定的被称为识别码的代码，对不同的级别进行识别（L 编码的二进制字，不考虑位的识别，将以符号<L>表示）。这个代码在编码字<L>以前被输送的是大量的零，因此，有可能识别紧随其后的游程长度 L 的值被编码的位数。L 的编码字用符号$\tilde{L}$表示，所以这个字由两部分组成：第一部分由零组成，使有可能识别 L 值所属的级别，第二部分<L>以 2 为底对这个值编码，具有这个编码所需要的最小位数。

下面的表格指示一个例子，值 m 等于 4，因此，L 可能有最大值 1023:

级别	分级编码(<L>)	识别码	分辨率
1－3	01－11	－	2
4－15	0100－1111	00	6
16－63	010000－111111	0000	10
65－255	01000000－1111111	000000	14
256－1023	0100000000－1111111111	00000000	18

不同的编码级 $Rn=2^{2n}$，$2^{2m+2}-1$ 是 1－3，4－15，16－63，64－255，

256－1023。“分级编码”列涉及对 L 值编码的二进制字＜L＞，这个 L 值表示符号 S 重复的数目。“识别码”列表示用以识别级别的那些位。“分辨率”列指示对 L 编码所需要的位数（这意味着完整的编码字γ_L将识别码和编码字＜L＞组合在一起）。

例如值 L＝5 被用 6 位二进制字（00.0101）编码，而值 49 则被用 10 位二进制字（0000.110001）编码。圆点是简单表示两个不同编码的邻接，即识别后面跟随＜L＞编码。由于不会有 L 值与以两个零开始的＜L＞编码字（L 至少等于 1）相对应，所以在编码中，对每一部分的识别或更精确地检测识别码，不会有什么问题。

让我们回到图 2 所表示的装置的功能。

与 5 级中的每一级相应的 5 个比较器 8，在它们的输入端并行接收 L 信息，每个比较器有三个输出，并根据所接收的输入值是否小于，处于或大于与这个比较器的级别属性相应的间隔，使第一，第二或第三输出起作用。这个信息向前送至逻辑电路 9，它的作用是产生与 L 信息所属级别相应的输出（这显然是一个实施例，也可以设想每个比较器的第二输出直接向前作为电路 6 的输出）。然后，级别检测电路以 5 个输出之一，向可变分辨率编码电路 7 输送与值 L 所属级别相应的信息。这个信息被电路 10 接收，所接收的每个 L 值作为编码电路 7 的输入。代码转换电路 10 所进行的代码转换，首先是截断由编码器 3 接收的 L－编码二进制字，获得＜L＞，从用于数值 1 至 1023 编码的 10 位的最初开始截留。这种截断有可能根据是否分别相应于第 1、第 2、第 3、第 4 或第 5 级的级别信息，而保留这个编码字的 2、4、6、8 或 10 个较低位。然后是在二进制字之前加上识别码，即根据是否分别相应于第 1、第 2、第 3、第 4 或第 5 级的级别信息，加上 0、2、4、6 或 8 位零值，给出 2、6、10、14 或 18 位的γ_L－编码字。代码转换后的二进制字被输送至多路复用器 11 的输入之一，多路复用器 11 在其输出端提供在其输入端出现的信息 L 的编码γ_L。因此，作为输出，编码器 3 提供一系列相应于它所接收的输入 L 值的γ_L编码字。

符号 S 的数值编码由符号编码器 4 进行。值 S 的编码原理详细表示于图 3。

编码器 4 的输入链接至字典 12 生成电路的输入，并接收符号 S 的二

进制值。电路的 7 个输出被链接至代码转换电路 13 的 7 个输入，代码转换电路 13 的输出就是符号编码器 4 的输出。编码器 4 的输入被链接至比较器 14 的第一输入和被称为字典 15 的随机存取存储器的第一输入。比较器 14 的输出被链接至 8 位计数器 16 的时钟输入。这个计数器 16 的输出被链接至随机存取存储器 15 的地址输入。它也被链接到逻辑电路 18 的第一输入和存储器电路 17 的输入，存储器电路 17 的输出被链接至逻辑电路 18 的第二输入。这个逻辑电路有 7 个输出，它们是字典 12 的生成电路的输出。它们被链接至代码转换电路 13 的 7 个输入，在代码转换电路 13 的输出端，提供符号 S 的编码，也就是$\tilde{s}$。

编码过程本质上是基于分配在字典中的地址。因此，字典 12 的生成电路每次接收到符号时，通过比较器 14，将符合的二进制与存储在字典 12 中的所有的值进行比较。为了这样做，图中未表示的管理电路对计数器进行控制，计数器每当新符号到达时上卷地址，作为字典 15 的输入，与这些地址相对应的每个存储在存储器中的值与所接收到的符号的二进制进行比较。当比较的结果为负时，8 位计数器通过与存储在存储器中的最新的地址比较而加 1，新的符号以这个新地址被存储在存储器中，并被加到字典中。

电路 17 存储最高地址，这个地址是存储字典中的符号所用的，并特别存储对这个地址编码“有意义”位的数目 t，也就是说最小数 t，使 2^{t-1} 大于或等于最高地址的二进制，t 的最小值被取为 2（所以值 t 能不减小）。t 值被向前输送至电路 18，使后者只发送位数，以相应于有意义位的数 t 那些较低位，对存储在存储器中的最新符号进行编码。这个逻辑电路的一个输出被分配给每个 t 值，对于存储 256 个值的字典来说，t 处于 2 至 8 之间。这些输出被输送至接着进行代码转换的编码电路 13。在第一输入端接收到的并起过两位的二进制信息一直向前直至最后输出。当二进制信息切换到第二输入端时，表示字典的地址以 3 位进行编码，在这个最新接收的 3 位二进制字的编码之前，是一串 10 位其值为零的码。二进制信息每次从一个输入端切换到另一个输入端时，表示编码以较多的位数进行，在这个输入端接收的第一个二进制字之前，加有一串 10 位其值为零的码。

字典以新地址进行地址（符号）编码。它只包含实用的那些符号，也

就是说只包含与OSD显示用的调色板中采用的颜色相应的那些地址。例如，其中符号以8位编码的一个部位，有256种颜色的一块调色板。如果在这个部位只有使用256种当中的64种颜色，那么，字典只包含64个符号。

所以，每个符号的编码是根据它自己在字典中的地址进行的。它用的是可变二进制分辨率编码，信息涉及这个通过加零而提供的分辨率。在前面的例子中有64种颜色和编码符号，具有6位的二进制分辨率而不是8位。单是通过这种操作，便可以节省25%。这是因为许多OSD部位使用远小于256种颜色，就使这种编码更为有效。

我们给出操作的例子如下:

在从位图数据中顺序阅读符号的循环开始时，假设OSD部位有少于4种颜色，因而被编码的第一符号具有2位的分辨率。每当一个新的未经编码的符号出现时，也就是字典中不存在这个符号时，它被存在字典的存储器中，并且地址加1。第一个符号被存储在地址0，所以，无论这第一个符号的值是多少，它的编码值是00。当一个新符号被读到时，它被存储在地址01，它的编码顺次为01。第四个新符号以同样的方法处理，它的编码是11。如果所读的其他值是那些存储在字典存储器中的值，它们的编码将是处于00和11之间的地址。那么，二进制分辨率就是2位而不是8位，所接收的符号就在2位上编码。这种情形一直到被接收的新符号不同于存储在字典存储器中的四个符号为止。然后，对于所遇到的所有其他符号，分辨率加1并转变为3位。如果读到的第6个符号已存在于字典的地址01，它的编码现在将是001。这个方法持续到字典包含多于128种颜色为止。从这个阶段开始，所有的符号具有8位分辨率。

为了进行符号译码而不致模糊，每当字典地址分辨率加1时，生成一个特定的代码。在最坏的情况下，这个代码将被生成6次。代码是一串10个零。这是因为只有9个顺序的零（包括8个用于识别码）是游程长度L（其值为255以上）编码所需要的。

对符号S编码的算法在图4中表示。步骤20是对不同参数进行初始化。在开始创建字典D时，没有数据存储在存储器中。

表示法:

符号S以形式C或Z.C.进行编码，

Z 是对应于 10 个零的识别编码，

C 对应于被存储在存储器中的符号二进制地址（位的数目可变），

t 是用来对 C 进行编码的位数，

i 是字典 D 的地址索引，被接收作为编码器输入的符号，以此地址存储在或回收到存储器，以及

n 是字典中存储器所存储的符号（或颜色）的数目。

初始化时，使 t 为 2，索引 i 被设置为地址 1（二进制形式为 01）。

当第一个符号 α 被接收时，它的值以字典的地址 1 存储在存储器中。

D（1）= α，n = 1。

在随后的步骤 21，一个新的符号被读出。

步骤 22 验证符号在字典 D 中是否已经存在。如果是，执行步骤 23，即搜索以字典地址 i 存储在存储器中对应于这个符号的字的值。下一步骤 24 是用 i 的二进制值对符号 S 进行编码。

C = 地址 i 的二进制形式。

如果符号在字典中不出现，则在步骤 22 之后运行步骤 25。这一步骤将值 n 加 1，并以地址 i = n 将新符号存入存储器。下一步骤 26 验证新值 n 是否需要附加的编码位。

因此，如果 n 大于值 2^t（其中 t 是最后进入字典的符号的地址所占用的编码位数），则进行步骤 27。由于被编码的新地址占用一个附加位，所以，值 t 加 1。因此，C 的二进制分辨率被加 1，10 个零的识别 Z 领先这个值 C。在步骤 27，进行如下操作：

生成 10 个零

t = t + 1

C 的二进制分辨率 = t

步骤 28 接在步骤 27 之后，在这一步骤中，索引 i 给定为值 n，符号 α 以地址 i 存入字典存储器。

i = n

D（i）= α

如果在步骤 26 指示 n 小于或等于值 2^t，则直接进入下一步骤 28。

步骤 28 之后是步骤 24，用地址 i 的二进制值对符号的值进行编码，

如果步骤 27 已完成，则以 10 个零领先。

图 5 表示根据图中参考号为 30 的位图数据流，进行编码的例子。L 对应于符号的重复数目，S 对应于符号的值。

$\tilde{S}$ 是对符号 S 编码的二进制字。在位图数据流 30 的编码步骤中，字典 D 用符号 S 的值填充，存储器的存储地址被用于对符号编码。要编码的值 32 是不同于前面的值的第五值，被存储在字典中。所以，根据数据流中的这个值，二进制编码字 $\tilde{S}$ 至少是 3 位的字。

$\tilde{L}$ 是对值 L 编码的二进制字。当 L 的值是在间隔 1－3 中时，二进制编码字 $\tilde{L}$ 是 2 位的字。当 L 的值是在间隔 4－15 时，二进制编码字 $\tilde{L}$ 是一个 6 位的字，前面两个零位指示编码的分辨率，因此 L 值占四位。

参考 31，被压缩的数据流表示在图的底部。它相应于每一游程长度的顺序二进制字 $\tilde{L}$ 和 $\tilde{S}$。每次在数据流 30 中遇到新的 S 值，则新的一对，S 被编码。在值为 32 的符号编码期间，二进制分辨率加 1，编码从这一点起以 3 位完成。因此，在二进制数据流中，在对应的一对 L，S 的编码之前，顺序的 10 个零被输送。

下面，对译码的方法和电路进行叙述。

由上述 OSD 编码器输出的被编码的位图数据，被存入视频译码器的 RAM 存储器，这个存储器被 OSD 编码器存取，另外，OSD 译码器被集成到这个视频译码器，如后面所述。存储在这个存储器的信息是颜色调色板，也是由视频译码器接收的规定 OSD 区域的坐标。也是由 OSD 编码器构成的并转送到 RAM 的字典。

根据本发明的译码装置 32 表示在图 6 中。

与视频译码器的 RAM 存储器联合的总线被链接到 FIFO（先进先出）存储器 33，以及具有上述字典容量的变换表或存储器 35。FIFO 存储器输出的数据被输送到字段识别电路 34。这个电路的第一输出端被输送到游程长度译码电路 37。第二输出端被连接到符号提取电路 36 的第一输入端。存储器 35 也接收从 RAM 存储器总线来的信息，它被连接到符号提取电路 36 的第二输入端。这后一个电路的输出被连接到用于生成符号序列的电路 38 的第二输入端。游程长度译码电路 37 的输出被连接到电路 38 的第一输入端，电路 38 的输出就是用虚线表示的译码器 32 的输出。

因此，一开始，来自由编码器为对 OSD 数据进行编码而生成的，并以变换表形式存储在 RAM 存储器的字典的数据，通过 RAM 存储器总线，转送到存储器 32，以便允许译码器在计算符号时，直接使用从这个存储器来的信息，这就是说，不需要存取 RAM 存储器。

点模式编码数据流（位图编码的位流或 BEB）或者被存储在视频译码器 RAM 存储器的被编码的位图数据，被输送到 FIFO 存储器。这个具有调整任务的存储器，将这些数据流输送到字段识别电路 31。这个电路能检测零序列，并识别编码字段。

FIFO 存储器根据上述压缩协议，提供由三个连续部分组成的数据流:

- 代码字段
- 游程长度字段
- 符号字段

这些字段的大小是可变的。具有最大为 19 位的代码字段对应于被插入的 10 个零序列，以指示 C（S 的编码）的二进制分辨率的变化，随后的是级别识别码（最大为 8 + 1 个零），指示 <L>（L 的编码）的二进制分辨率。因此，有可能确定对 L 的值进行编码所占用的位数 n_i，以及对符号的地址进行编码所占用的位数 n_s。

当代码字段零的个数小于 10 时，相应的零序列例如 p 个零，仅是在这个数目高于 10 时才涉及 L 的编码，前 10 个零指示 C 的二进制分辨率加 1，序列中其余 p 个零对应于 L 的编码。

p 值指示对游程长度进行编码占用的位数 n_i:

- 如果 p 是偶数，$n_i = p + 2$
- 如果 p 是奇数，$n_i = p$。

n_i 和 n_s 的值一旦计算完毕，符号和游程长度字段能被辨别。对应于这些字段的值能被提取。

因此，字段识别电路 34，从与游程长度编码相对应的数据流中提取被编码的二进制字，以便将它们从第一输出端输送游程长度译码电路 37，并且提取对应于符号编码数据流的被编码的二进制字，以便将它们从第二输出端输送到符号提取电路 36。游程长度译码电路在它的输出端提供 L 值。符号提取电路根据从字段识别电路 34 接收的地址，寻找存储器 35 中以这

个地址所存储的符号 S 的二进制值。用于生成符号序列的电路 38 接收 L 和 S 值，在它的输出端给出相应的每个 S 重复 L 次的数据流。

这个 OSD 译码器所采用的译码算法表示在图 7 中。

第一步骤 40 是读出存储在视频译码器 RAM 存储器的字典，并将其转送至 OSD 译码器的存储器 32。用于对 C 编码的位数 t 被初始化为 2。下一步骤 41 检测由 FIFO 存储器送来的数据流中的零序列，只要这个相继零的个数 n 小于 10，就将 n 存储到存储器。这个数一旦等于 10 或者这个零序列一旦被终止，则启动步骤 42，它将区分这两种情况:

- 如果 n 已经达到值 10，则后一步骤 43 将 C 编码用的 t 值加 1，然后回到步骤 41，并且测量在前面 10 个零被检测之后，n 剩余的新的零的个数。

- 如果 n 小于 10，步骤 44 对游程长度进行译码。为此，小于 10 的零的个数，或者如果经过步骤 43 之后剩余的零的个数，即前面称为数 p 被使用来确定相继于这些零的位数，被考虑用作对 L 的译码，这个数是 p 的奇偶函数，如上所指出的。

在对游程长度 L 译码之后，下一步骤 45 读取跟在游程长度编码位数后面的位数 t，以确定 C 值。步骤 46 搜索存储器 32 中地址 C 的内容即 S 值。

图 8 给出根据本发明已编码的数据流进行译码的例子，图中数据流的参考号是 47。

- 开始，值 n_s 初始化为 2。代码字段对应于单个零。因此，n_s 值保持等于 2，n_i 值等于 1（p=1）。游程长度字段包含的值为 1，符号字段包含的值为 00。因此，L 的值等于 1，S 的值是地址为 00 的字典的内容，即为 6。下一代码字段包含单个零值，因此，字段的长度不变，符号的值在字典的地址 01 被找到。当相继的零比较多或者等于 10 时，对 L 编码的二进制分辨率就要改变。在我们的例子中，出现相继 13 个零。因此，n_s 被加 1，读出 S 用 3 位。这三个继 10 位零之后的零，指示游程长度字段具有三位的尺度。因此，L 用 3 位编码为 101，接着 S 用 3 位编码为 100。所以，在地址 100 找到值 32 被重复 5 次。

由此所得到的译码数据流被表示在图的底部，参考号为 48。它对应于

视频译码器所接收的顺序为 S 的二进制字，也就是说，在 OSD 编码之前，存储在 RAM 存储器的二进制字。

图 9 给出一个由 OSD 编码器和 OSD 译码器组成的编码译码器的结构例子，它们安装在视频译码器中。图中用虚线表示的编码译码器 50 通过总线被连接到 RAM 存储器 49。

位图数据，OSD 区域的坐标和颜色调色板，视频译码器所接收的数据以及有关 OSD 的信息被输送到 OSD 编码译码器的输入端，例如通过视频译码器的微控制器接口，在图中未表示。这个输入被连接到总线控制器电路 51 的输入端，OSD 信息通过总线控制器电路 51，总线控制器电路 51 将位图数据，通过第一输出端，输送到 OSD 编码器 52 和颜色调色板以及 OSD 坐标；通过第二输出端，输送到输入／输出电路 53 的第一输入端。电路 53 采用双向连接方法，与编码译码器外部的装置进行数据交换，特别是与 RAM 存储器。电路 53 通过第一输出端被连接到 OSD 译码器 54，并通过第二输出端被连接到编码译码器 50 的第一输出端。OSD 译码器 54 的输出端被连接到编码译码器 50 的第二输出端。这个编码译码器 50 的第二输出端被连接到存储器 55 的第二输入端。最后，存储器 55 被连接到生成亮度和色度信号的电路 56，该电路提供 Y、U、V 信息作为输出。

因此，由视频译码器所接收的位图数据通过总线控制器 51 被输出到 OSD 编码器 52，而在这个编码器输出端，已编码的数据通过输入／输出电路 53 被输送到视频译码器的 RAM 存储器，被存储在这里。其他的有关调色板和坐标的 OSD 数据，通过总线控制器和输入／输出接口也被输送到 RAM 存储器，将被存储在存储器各区域中。由编码器创建的字典也被传送到 RAM 存储器。所有这些信息，位图数据被压缩，调色板，坐标和字典的内容则不被压缩，这些信息由 OSD 译码器 54 使用。译码器 54 通过输入／输出接口 53 接收被编码的位图数据流或者 BEB（位图被编码的位流），以及字典。这些数据由 OSD 译码器 54 进行译码，将其输出到 CLUT 存储器 55，并且，对于每个像素 i，输送一个地址（未编码），这个地址是含有与这个像素相应的颜色的调色板地址。这个存储在 RAM 存储器的调色板，在对位图数据译码之前，通过输入／输出接口 53 已经输送到存储器 55。数字亮度和色度数据作为输出，从存储器 55 被输送到显示电路（显示单元）56，显示

电路同时也接收视频信息。

编码译码器已经叙述的一种情况是，OSD 信息是被输送到译码器的。在这些信息是常驻的情况下，常驻的位图数据可以是别处被编码的位图数据，则不需要在视频译码器中安装 OSD 编码器。所以，对于本发明的功能来说，在视频译码器。所以，对于本发明的功能来说，在视频译码器中只有 OSD 译码器是必要的。

本发明的 MPEG II 视频译码器的文本中已经被叙述，很明显，这里涉及到的是一个实施例，这个编解码器在任何类型的译码器、电视机、视频录像机或其他装置中的安装将不会脱离本发明的范围。

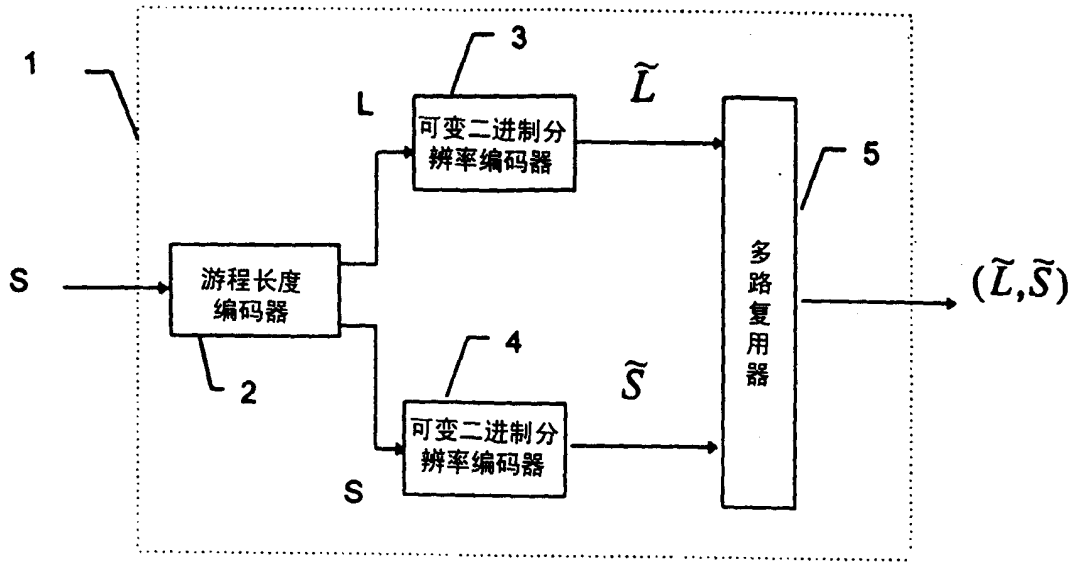

1
3
L
$\tilde{L}$
可变二进制分辨率编码器
5
S
游程长度编码器
多路复用器
$(\tilde{L},\tilde{S})$
2
4
$\tilde{S}$
可变二进制分辨率编码器
S

图 1

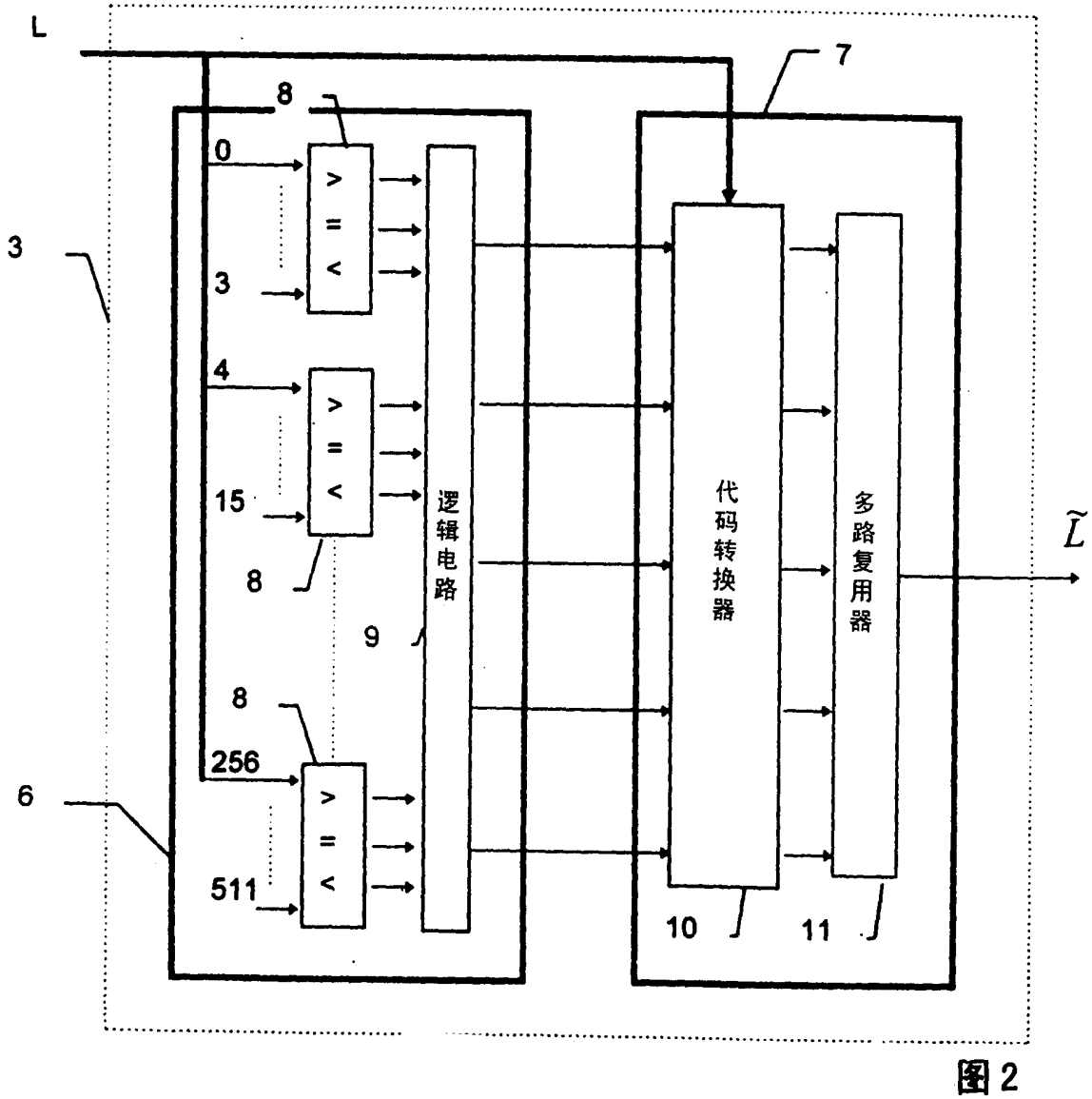

L
7
8
0
3
>
=
<
3
4
15
8
逻辑电路
9
代码转换器
多路复用器
$\tilde{L}$
8
256
511
6
10
11

图 2

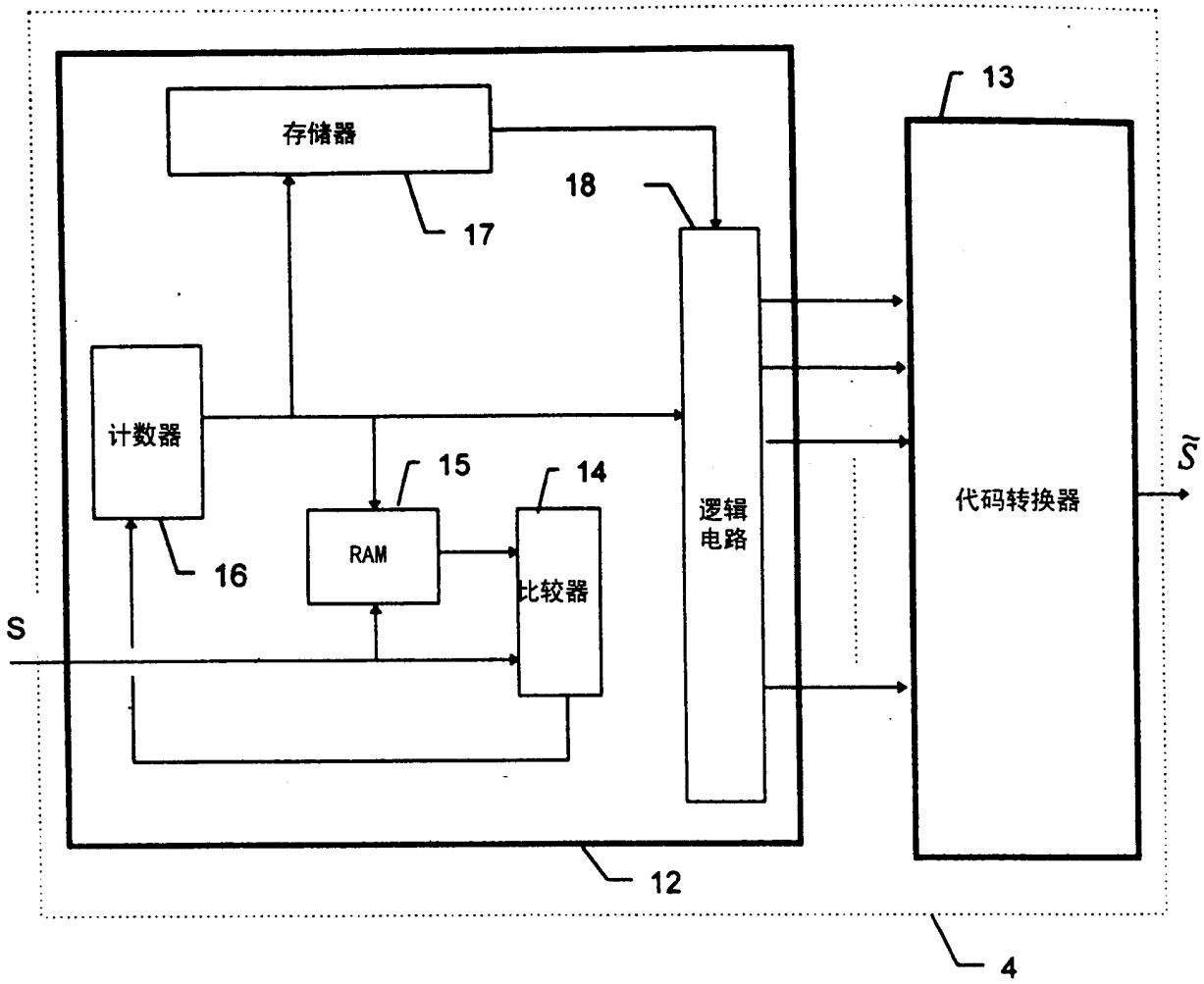


图 3

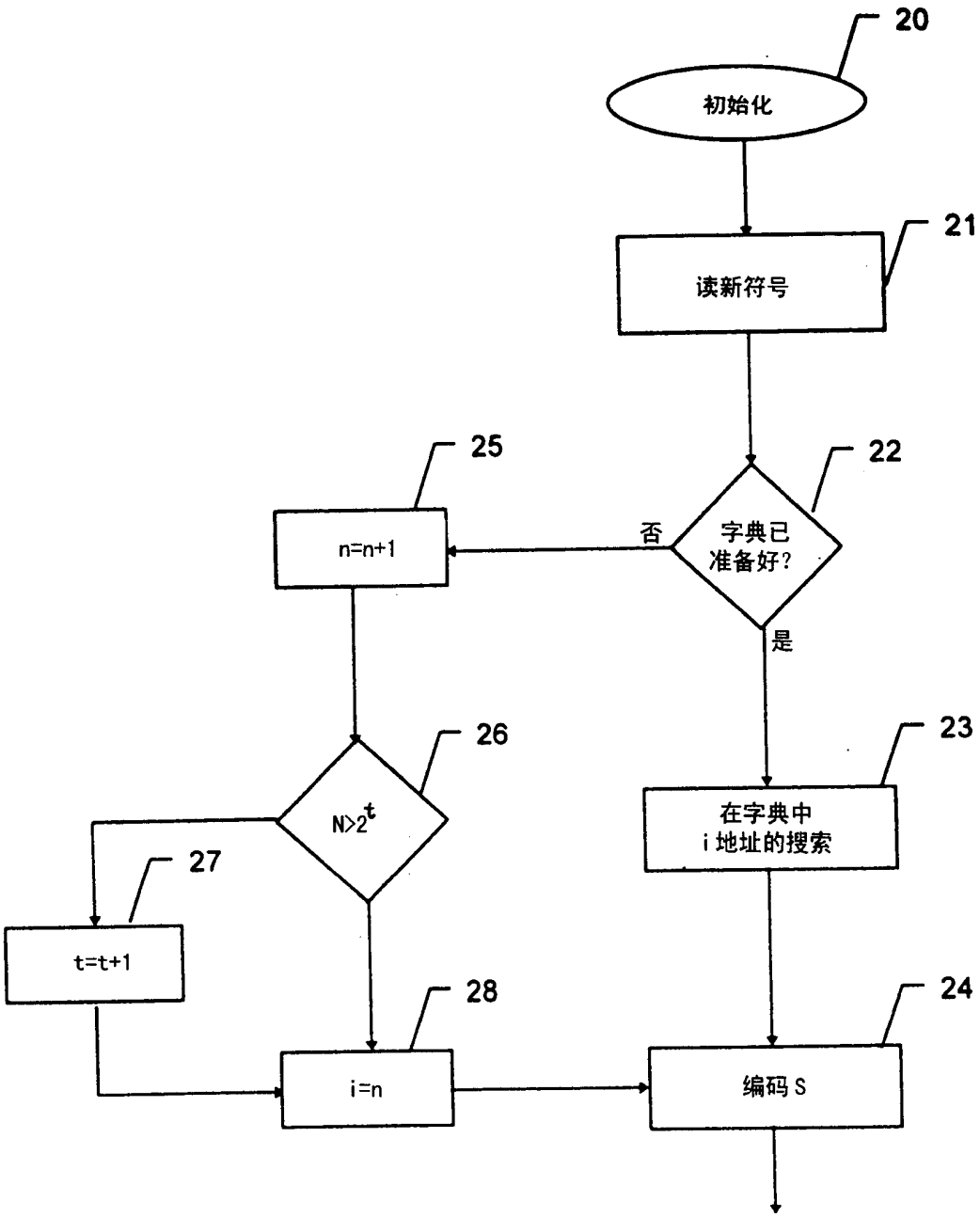

20
初始化
21
读新符号
22
字典已
准备好?
否
是
25
n=n+1
26
N>2^t
27
t=t+1
28
i=n
23
在字典中
i 地址的搜索
24
编码 S

图 4

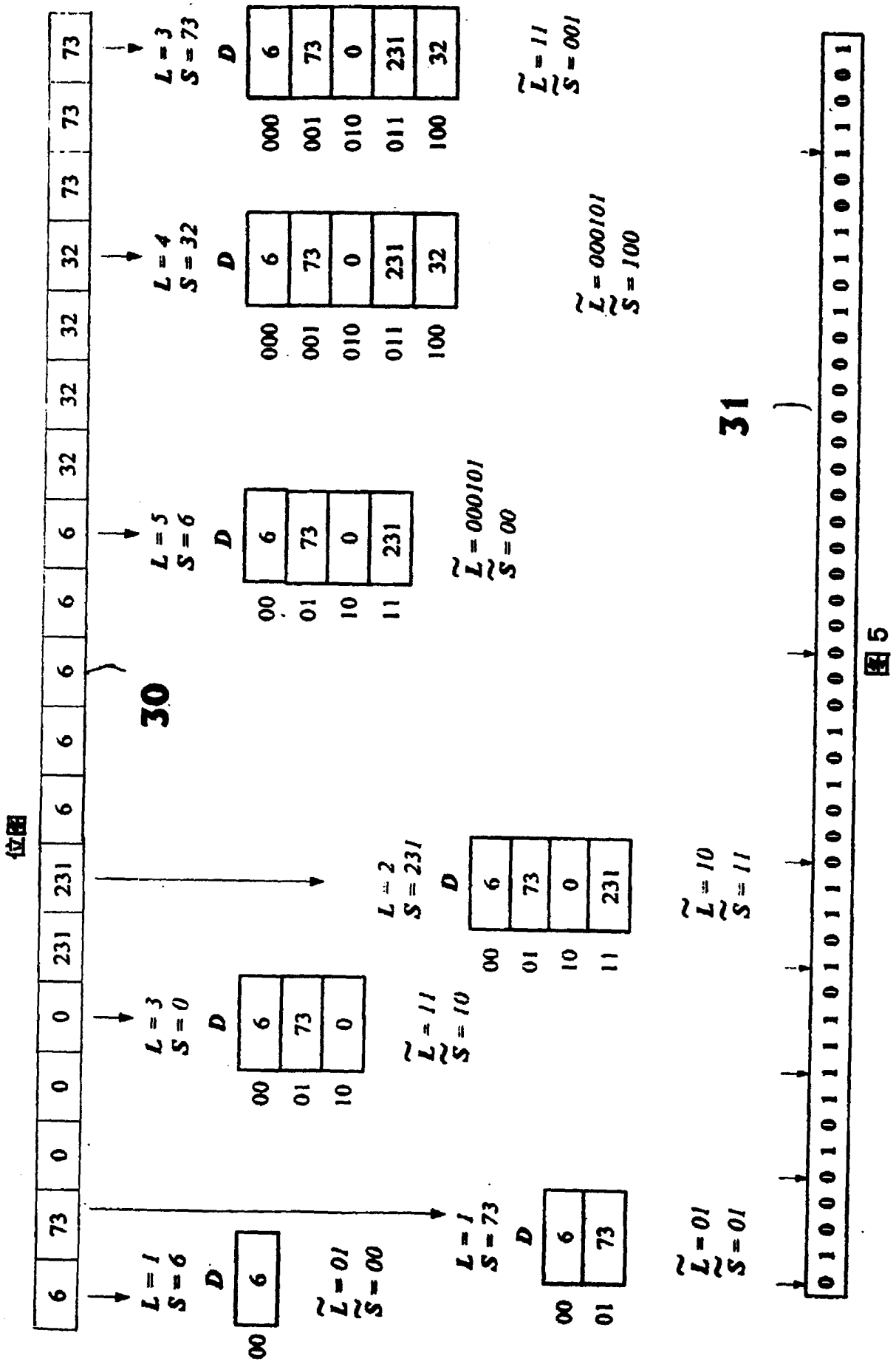

位图
6 73 0 0 0 231 231 6 6 6 6 32 32 32 32 73 73 73
30
L = 1
S = 6
D
6
00
L = 3
S = 0
D
6 73 0
00 01 10
L = 5
S = 6
D
6 73 0 231
00 01 10 11
L = 4
S = 32
D
6 73 0 231 32
000 001 010 011 100
L = 3
S = 73
D
6 73 0 231 32
000 001 010 011 100
L = 2
S = 231
D
6 73 0 231
00 01 10 11
L = 1
S = 73
D
6 73
00 01
~L = 01
~~S = 00
~L = 11
~~S = 10
~L = 000101
~~S = 00
~L = 000101
~~S = 100
~L = 11
~~S = 001
~L = 10
~~S = 11
~L = 01
~~S = 01
31
0 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1

图5

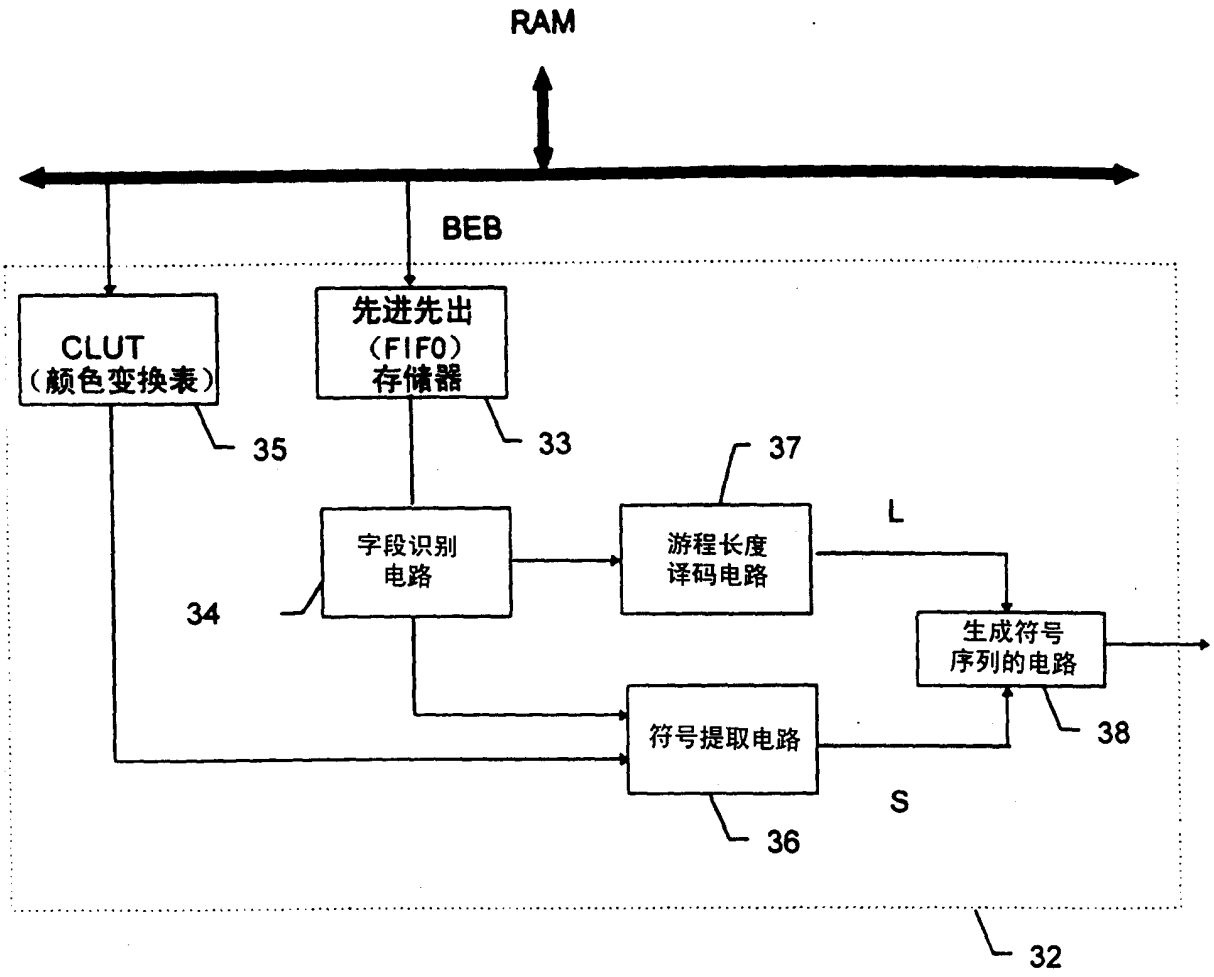

RAM
BEB
CLUT
（颜色变换表）
35
先进先出
（FIFO）
存储器
33
37
字段识别
电路
34
游程长度
译码电路
L
生成符号
序列的电路
38
符号提取电路
S
36
32

图 6

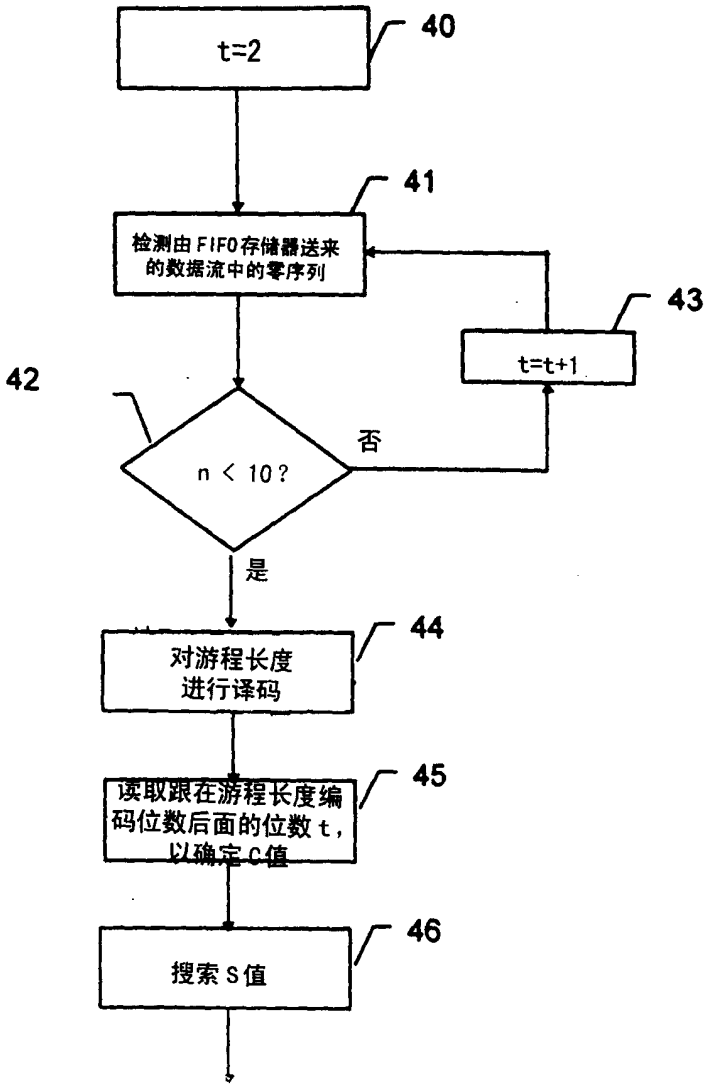

t=2
40
41
检测由FIFO存储器送来的数据流中的零序列
43
t=t+1
42
n < 10?
否
是
44
对游程长度进行译码
45
读取跟在游程长度编码位数后面的位数t，以确定C值
46
搜索S值

图 7

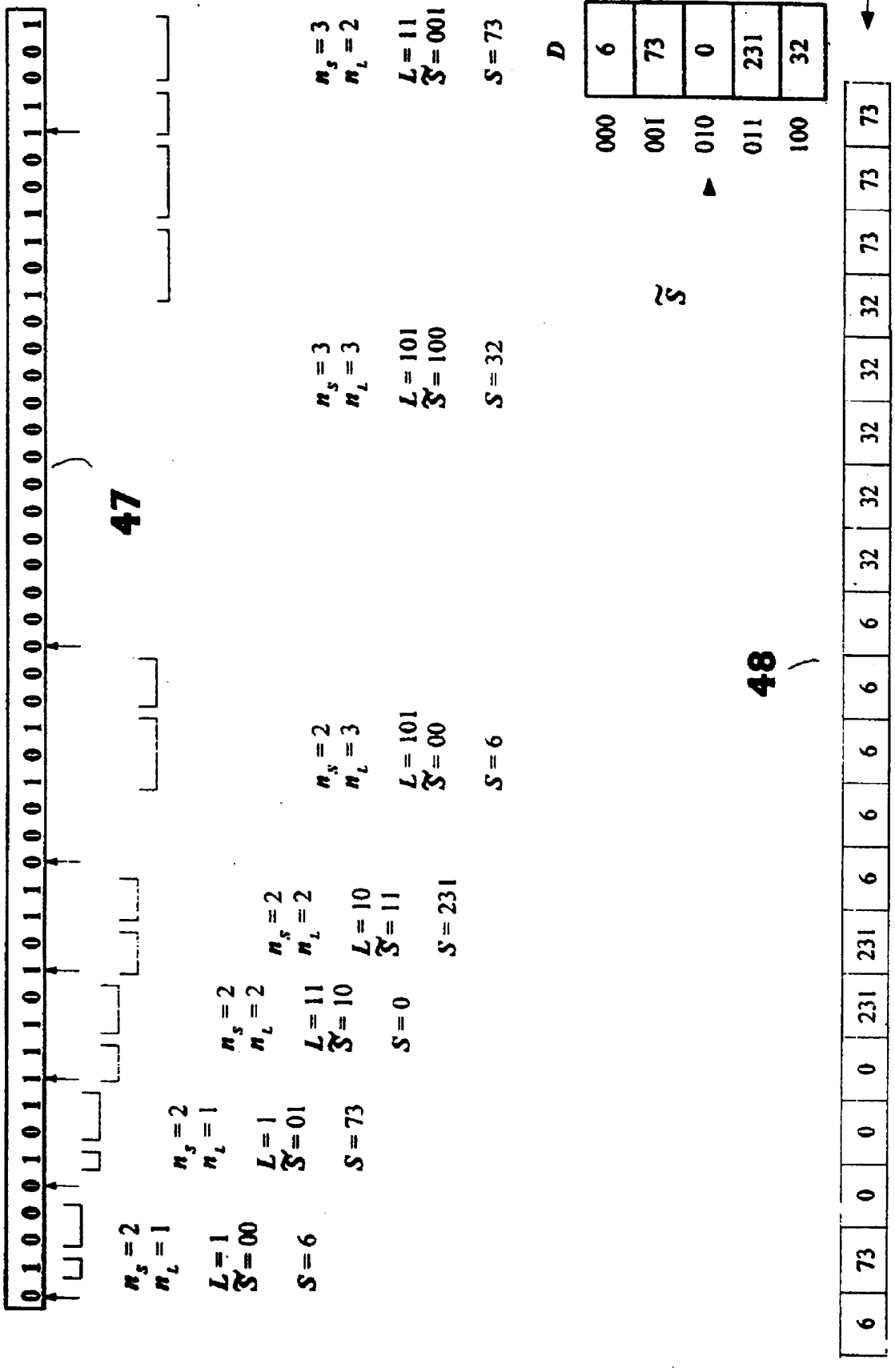

0 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1
47
$n_s = 2$ $n_L = 1$ $L = 1$ $\tilde{S} = 00$ $S = 6$
$n_s = 2$ $n_L = 1$ $L = 1$ $\tilde{S} = 01$ $S = 73$
$n_s = 2$ $n_L = 2$ $L = 11$ $\tilde{S} = 10$ $S = 0$
$n_s = 2$ $n_L = 2$ $L = 10$ $\tilde{S} = 11$ $S = 231$
$n_s = 2$ $n_L = 3$ $L = 101$ $\tilde{S} = 00$ $S = 6$
$n_s = 3$ $n_L = 3$ $L = 101$ $\tilde{S} = 100$ $S = 32$
$n_s = 3$ $n_L = 2$ $L = 11$ $\tilde{S} = 001$ $S = 73$
D
S
000 001 010 011 100
6 73 0 231 32
$\tilde{S}$
48
6 73 0 0 0 231 231 6 6 6 6 32 32 32 32 32 73 73 73

图 8

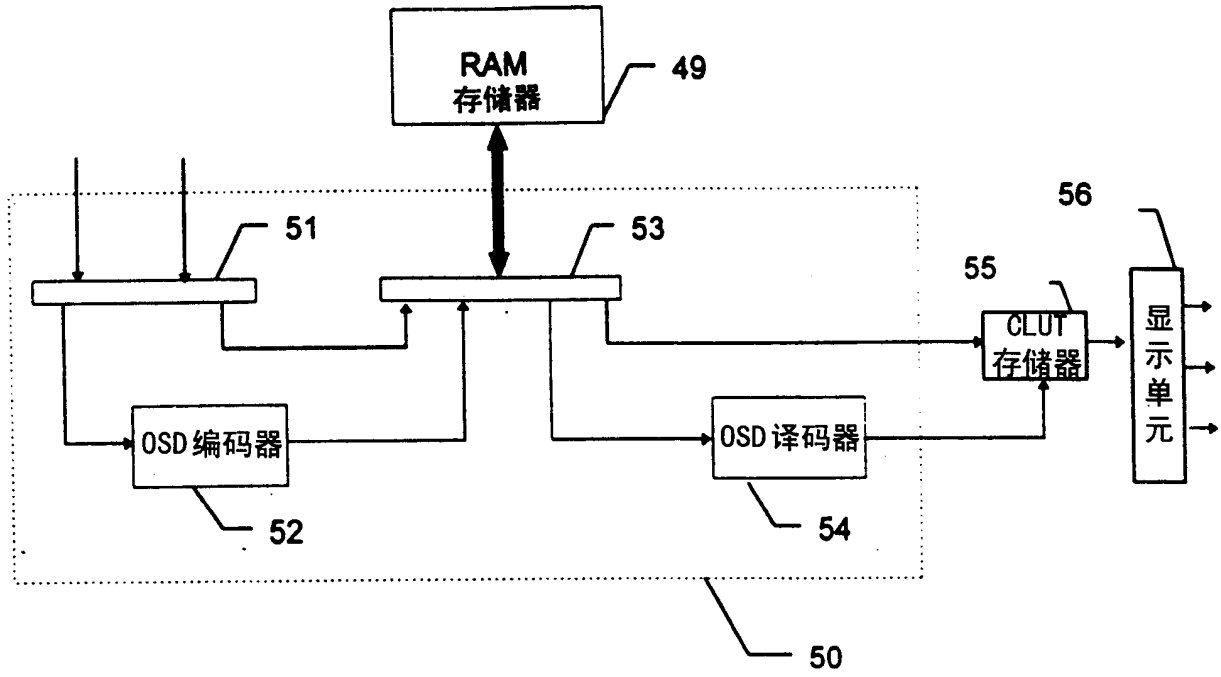


图 9