



# [12] 发明专利说明书

专利号 ZL 99807908.1

[45] 授权公告日 2007 年 6 月 27 日

[11] 授权公告号 CN 1323402C

[22] 申请日 1999.7.1 [21] 申请号 99807908.1

[30] 优先权

[32] 1998. 7. 13 [33] US [31] 09/115,530

[86] 国际申请 PCT/GB1999/002078 1999. 7. 1

[87] 国际公布 WO2000/003392 英 2000. 1. 20

[85] 进入国家阶段日期 2000. 12. 26

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 马丁·A·哈斯那 尼勒斯·海斯

沃尔特·赫特 巴里·M·特拉格

[56] 参考文献

US5774078 1998. 6. 30

审查员 郭 雯

[74] 专利代理机构 中国国际贸易促进委员会专利

商标事务所

代理人 王以平

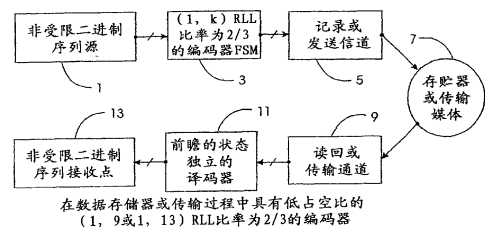
权利要求书 2 页 说明书 9 页 附图 12 页

## [54] 发明名称

将二进制序列可逆变换为比率为  $2/3$  (1, K) 游程受限的编码序列

## [57] 摘要

一种通过将非受限二进制信号序列逆映射到一个 (1, k) 比率  $2/3$  的 RLL 的方法和装置来抑制在存储和通信过程中的非受限二进制信号序列的高占空度, 使高占空比大大地被抑制到大约  $1/3$  左右。也就是, 可以防止或消除普通的映射到高占空比的 RLL 码序列的二进制序列存在的不确定的重复中。



1.一种将二进制序列逆映射到一个比率  $2/3(1, k)$ 的具有最大传输密度约束的游程受限码序列 RLL 的方法, 该方法包括步骤:

(a) 在一个处理器内定义并存储包含下一个状态 $(n_1n_2n_3)$ 和一个当前 RLL 代码的三位符号 $(c_1c_2c_3)$ 的有序对的状态转换表, 根据其当前状态做为第一个表格维, 和根据当前 $(b_1b_2)$ 位对和预定数目的二进制序列的前瞻 $(b_3b_4)$ 位对的矢量 $(b_1b_2b_3b_4)$ 做为第二表格维, 索引所述转换表中的每一个有序对, 每一个当前的位对都是一个可识别的二进制值 $(00, 01, 10, 11)$ , 从一组由可识别二进制值和一个无关 $(xx)$ 数值组成的集合中选择每个预定数目的前瞻位对, 状态对状态的转换受到约束, 使得 RLL 编码三位的任何一个对应的长期序列表明一个占空比是小于 50%的;

(b) 响应一串连续的位对矢量使上述的处理器存取该表, 并从中提取一串连续的 RLL 编码三位 $(c_1, c_2, c_3)$ 和一连串连续的内部状态 $(n_1, n_2, n_3)$ ; 以及

(c) 响应一串连续的 RLL 编码三位 $(c_1, c_2, c_3)$ 使所述处理器存取该表, 并从中提取一串连续的二进制序列位对。

2.一种如权利要求 1 中的方法, 其中所述占空比实质上近似为三分之一, 更进一步, 其中所述的 $(1, k)$ RLL 码约束是从包含 $(1, 9)$ ,  $(1, 10)$ 和 $(1, 13)$ 中选择的。

3.一种如权利要求 1 中的方法, 其中所述的 $(1, k)$ RLL 码约束是从包含 $(1, 9)$ 和 $(1, 13)$ 中选择的。

4.一种将二进制序列逆映射到一个比率  $2/3(1, 9)$ 的具有最大传输密度约束的游程受限码序列 RLL 的设备, 该设备包括:

一个有限状态机 FSM, 用于产生一个 RLL 编码的三位符号 $(c_1c_2c_3)$ 和下一串连续的内部机器状态 $(n_1n_2n_3)$ , 该 RLL 编码的三位符号 $(c_1c_2c_3)$ 作为当前的一组 FSM 的内部状态 $(s_1s_2s_3)$ 和一个当前 $(b_1b_2)$ 位对和一个来自二进制序列的预定数目的前瞻 $(b_3b_4)$ 位对的矢量 $(b_1b_2b_3b_4)$ 的第一

函数，该串连续的内部机器状态( $n_1n_2n_3$ )作为当前 FSM 状态和该二进制序列向量的第二函数，每一个当前的位对成为一个可识别的二进制值(00, 01, 10, 11)，每一个预定数目的位对是从由包含一个可识别二进制值和一个无关值(xx)的一个集中选取的，所述的第一和第二函数由下列的一组布尔关系式确定：

$$n_1 = b_1s_3 + b_1b_2'b_3's_1's_2 + s_1s_3$$

$$n_2 = b_1's_3$$

$$n_3 = b_2's_3 + 's_1'b_1b_2$$

$$c_1 = 's_1s_2$$

$$c_2 = 's_1's_2'c_3$$

$$c_3 = 's_1s_3('b_1 + 'b_2) + 's_1's_2's_3b_1b_2'b_3b_4.$$

## 将二进制序列可逆变换为比率为 2/3(1,k)游程受限的编码序列

### 技术领域

本发明涉及在例如磁存储记录信道或光通信子系统中出现的二进制值序列的非正常译码，其中误码、擦除或故障包含具有高占空比码型或所选码型重复的二进制子序列。更具体地，本发明涉及一种防止或消除码型的不确定的重复的方法和装置。

### 背景技术

众所周知，由于其中存在非设定的码型或属性，一个二进制值序列如 1000011011 可能被译码设备错误地译码。例如，一个 1010101010...序列具有很高的占空比。从电子学上看，一个高占空比的意义等同于电子或机械元件被高度重复地动作。这个高频动作产生伴随发热并增加在元件环境中的噪声，从而导致较高误码率和故障率，并缩短元件寿命。

为此定义了占空比这个术语，它的意义是：一个码型或一个重复子序列间隔中出现二进制数字 1 的数量。在 101010 这个例子中，因为每两位间隔就有一个 1 出现，所以它的占空比是 50%。如果一个重复码型具有两个二进制 4 位字的形式，如 1001, 1000，则占空比就是  $3/8=37.5\%$ 。

在磁记录信道和光通信中存在脉冲拖尾或展宽现象。例如，两个二进制 4 位字 0001, 0010，由于拖尾，在译码器的输入处呈现为 0001, 1010。在这种情况下，位于第一个字的第 4 位的二进制“1”被电子地或光学地延伸，就象存在两个二进制“1”一样。这种情况或拖尾在脉冲位置调制通信系统或类似系统特别多。由此导致了 PPM 系统的明显的带宽效率极低的缺点。

从 1983 年 11 月 1 日授权给 Adler 等人的标题是“用于产生具有比率为 2/3 的(1, 7)信道的无噪声滑动块代码的方法和设备”的美国专

利 4, 413, 251 可以得知, 有限状态机(FSM)能将非受限二进制值序列以不可逆方式转换到一个受限二进制值序列。进一步, Adler 讲述了一种有限前瞻独立状态机可以执行这种译码。前瞻能力使一个译码器通过考虑一个预定数目的子序列 RLL 码字, 将一个当前的 RLL 码字的  $n < m$  位分解为  $m$  位非受限序列。尽管如此, 这种译码器的前瞻特点仍会增加 RLL 码字的误码或消除的不利影响。

参照 Adler 等人的方法, 对于将  $m$  个位的非受限二进制序列映象到  $n$  个位的受限二进制序列的给定码率  $R=m/n$ , 该要求通过得到一个每个状态具有  $2^m$  分支的 FSM 编码器被部分满足, 在每个状态内, 通过拆分和合并某些 FSM 状态以得到一个新的 FSM 来表明  $(d, k)$  约束。该  $(d, k)$  约束的含义是在任何一对连续的二进制“1”之间插入至少  $d$  个“0”, 且至多不超过  $k$  个“0”。对于  $d < k$ ,  $d$  就确定了转变频率和符号间干扰(ISI)的来源。而  $k$  则用于时钟的再同步。

#### 发明内容

本发明提供一个用于将非受限二进制序列转换到一个受限序列的 FSM, 该受限序列是从一系列固定比率为  $2/3(d, k)$  的由  $(1, 9)$  和  $(1, 13)$  RLL 代码组成的 RLL 代码中选择的, 其中与  $(1, 9)$  和  $(1, 10)$  RLL 代码的情况相同, 抑制预先确定的 RLL 代码序列不确定的重复出现, 或与  $(1, 13)$  RLL 代码的情况相同, 消除预先确定的 RLL 代码序列不确定的重复出现。更进一步, 当从一个存储子系统或一个光通信路径或类似的路径中读取 RLL 代码序列时, 一个前瞻的非基于状态译码器提供了必须的可逆性。

更实际地, 通过用于将二进制序列逆映象到一个具有最大转移密度限制的比率为  $2/3(1, k)$  的游程长度受限码(RLL)序列的一个处理器实现方法, 或一个硬连线组合逻辑等价物, 确信可以实现前述目的。这种方法包括两个步骤, 也就是, 定义和存储一个有限状态机作为一个编码器, 然后对二进制序列执行映象。

第一步骤包括在一个处理器内定义并存储包含下一个状态  $(n_1 n_2 n_3)$  和一个当前 RLL 代码的三位符号  $(c_1 c_2 c_3)$  的有序对的状态转换表。根据其当前状态在预定数目的第一个表格位置, 并根据二进制序列的当前

$(b_1, b_2)$ 位对和前瞻 $(b_3, b_4)$ 位对的向量 $(b_1b_2b_3b_4)$ 在第二个表格位置,索引所述表中的每一个有序对。每一个当前的位对都是一个可识别的二进制值(00, 01, 10, 11)。而且从一组由可识别二进制值和一个“无关”的(xx)值组成的集合中,选择每个预定数目的位对。有意义的是,状态对状态的转换是受限制的,因此 RLL 编码三位的任何对应的长期序列表明占空比小于 50%。第二步骤包含使处理器响应一个后继的位对矢量存取该表,并从中提取一个后继的 RLL 编码三位。在图及优选的实施例中,详述了定义 RLL 代码字到二进制的映象的逻辑关系以及其相反逻辑关系。

为了使用一个 $(1, k)$ 比率  $2/3$  的 RLL 编码来减少二进制值流的占空比,有必要理解当将所选择的非受限二进制编码码型用于一个标准 $(1, 7)$ 或 $(1, 9)$ 比率  $2/3$  的 RLL 编码器时可能会发生的功效。的确,这些所选择的码型所产生的受限二进制编码码型具有很高的占空比。举例参见下面的表 1。

表 1

非受限二进制编码码型	$(1,7)$ 比率 $2/3$ RLL 编码	占空比
00 11 00	010 101 010	4/9
00 11 01	010 101 001	4/9
	$(1,7), (1,9)$ 比率 $2/3$ RLL 编码	
10 11 00	100 101 010	4/9
10 11 01	100 101 001	4/9
	$(1,7), (1,9)$ 比率 $2/3$ RLL 编码	
00 11 10 11	010 101 000 101	5/12
10 11 10 11	100 101 000 101	5/12

但是,更进一步观察可知,Adler'413'专利公开的 $(1, 9)$ 和 $(1, 13)$ 比率为  $2/3$  的 RLL 编码器类型,可以启发地改进为(a)或(b), (a)为在这些预定的非受限码型中禁止无限的高占空比 RLL 编码的重复, (b)为有效地禁止这些重复,如下表 2 所示。

表 2

非受限二进制编码码型	带修改的编码器/译码器 的(1,9)比率 2/3RLL
00 11 00	010 000 000
00 11 01	001 000 000
	带修改的编码器/译码器 的(1,13)比率 2/3RLL
10 11 00	100 000 000
10 11 01	101 000 000
00 11 10 11	010 000 000 000
10 11 10 11	100 000 000 000

附图说明

图 1 示出了一个根据本发明的定位于关于信息存储和传输媒体的受限信道编码器和译码器；

图 2 给出了一个用做一个编码器的表示输出是当前内部状态和一组输入的函数，以及下一个内部状态是当前内部状态和一组输入的另一函数的有限状态机；

图 3 描述了一个响应非受限二进制输入产生高占空比输出的状态图(1, 7)比率为 2/3 的 RLL 编码器，其中一个状态修正将该编码器转换为一个(1, 9)比率为 2/3 的产生低占空比的输出的 RLL 编码器；

图 4A-4C 分别说明了根据本发明修改的一个用于(1, 9)比率为 2/3 的 RLL 编码器的状态转换表，涉及此非受限输入到受限输出的布尔运算及该编码器的一个高级逻辑门的实施例；

图 5A-5C 分别示出了根据本发明修改的一个用于(1, 9)比率为 2/3 的 RLL 译码器的状态转换表，涉及此非受限输入到受限输出的布尔运算及该译码器的一个高级逻辑门的实施例；

图 6A-6C 分别描述了根据本发明修改的一个(1, 13)比率为 2/3 的 RLL 编码器的涉及非受限输入到受限输出的布尔运算，一个状态转换表，以及该编码器的高级逻辑门的实施例；和

图 7A-7C 分别陈述了根据本发明修改的一个用于(1,13)比率为 2/3 的 RLL 译码器的译码表,涉及受限输入到非受限输出的布尔运算,以及该译码器的高级逻辑门的实施例。

### 具体实施方式

现在参照图 1,该图示出了一个位于一个记录或发送信道 5 及一个信息存储器或传输媒体 7 之前的受限信道编码器 3。一个与之对应的译码器 11 位于一个读回或接收信道 9 和非受限二进制序列接收点 13 之间。二进制源 1 可以看作一个具有按照一个一致的,高斯或其他统计分布并常常展示具有高占空比的重复码型的“1”或“0”的随机数发生器。这些占空比的范围从具有 10101010...等形式的最高 50% 的序列到一些可容许远远小于 50% 的典型序列。因为通常的解决方法是将非受限二进制序列应用到一个(1, k)比率为 2/3 的 RLL 编码器,该编码器自身提供最小占空比。

接着,来自编码器 3 的受限二进制输出字符串,被用于一个记录或传输信道 5,在这里被适当地调制并写入到一个存储器或一个传输媒体 7 中。在后续的时间里,受限并调制的字符串到达一个读回或一个接收信道 9 中,并被转换成一个受限的由“1”和“0”组成的二进制字符串。该被转换二进制字符串接着被送到前瞻的状态独立的译码器 11 中。译码器 11 利用预定数目的后续字符(前瞻的)并将约束除去,然后将原始的非受限二进制序列提供到接收点 13。

现在参照图 2,图 2 展示了一个用做一个编码器 3 的一个有限状态机(FSM)。对于一组特定的参数(d, k)=(1, 9)或(1, 13),预定数目的当前输入位( $b_1b_2$ )和将来输入位( $b_3b_4$ )的( $b_1b_2b_3b_4$ ),从输入寄存器 21 通过路径 22 被送到 FSM23。FSM23 提供一个输出到寄存器 25。作为输入位( $b_1b_2b_3b_4$ )和当前机器状态  $S=(s_1s_2s_3)$  的函数  $f$ ,得到一个二进制位  $c=(c_1c_2c_3)$  的输出集合。同时,下一个机器状态  $N$  可以由输入位 ( $b_1b_2b_3b_4$ )和当前机器状态  $S$  的函数来确定。因此,对于下一个周期,下一个机器状态  $N$  被写入到当前状态寄存器 29 中,成为新的当前机器状态  $S$ 。就这点而言,一个 FSM 也可以被看作是一种有向图的形式,

该图的节点代表了 FSM 的机器状态，且它的有向连接在产生一个对应的输出矢量的同时，定义任一给定输入矢量的状态到状态的转换。有向图的表格形式被称为“状态转换表”。

现在参照图 3，图 3 描述了响应一个非受限二进制输入产生高占空比输出的(1, 7)比率为 2/3 的 RLL 编码器的部分状态图或有向图。节点 000, 001, 010, 011 和 100 代表这个(1, 7)编码器的内部状态。因为这是一个比率为 2/3 的编码器，所以对于一个在时间 001100 具有两个位的输入码型，编码器的输出码型将是在时间 010101010 具有三个位的输出码型。类似地，对于输入码型 001101，输出码型将是 010101001。如图所示，如果一个附加状态 111 被加到状态图中，并连接在状态 000 和 100 之间，该编码器就被转换成(d, k)=(1, 9)，比率为 2/3 的编码器。该附加状态提供足够的计算资源，来前瞻几个输入符号并产生一个低占空比的输出序列，如对于输入 001100 的输出序列为 01000000，对于输入 001101 的输出序列为 00100000。

现在参照图 4A-4C，图 4A-4C 分别示出了根据本发明修改的一个用于(1, 9)比率为 2/3 的 RLL 编码器的状态转换表，涉及这个非受限二进制输入到高度受限的输出的布尔运算及该编码器的一个高级逻辑门的实施例。在图 4A 中，编码器是一个具有六个内部状态 S 的 FSM。当该编码器的比率为 2/3 时，它被设计用于响应一个当前输入位对加一个前瞻输入位对  $b=(b_1b_2b_3b_4)$ 。每一行由六个内部状态之一来索引，而每一列则由具有两个位对组合的七个输入之一来索引。对于任何给定(S, b)的表索引，入口形式为下一个状态 N/输出  $C=(c_1c_2c_3)$ 。在七个具有两位对输入组合的输入中，其中的三个，即，00xx, 01xx 和 10xx，有两个“无关”的位值位置。举例说明如下，如果该(1, 9)编码器在状态 100 的输入是 00xx，那么将产生一个从状态 100 到状态 000 的传输且产生输出  $C=000$ 。

现在参照图 4B，图 4B 陈述了定义下一个状态的布尔函数  $N=(n_1n_2n_3)=g(S, b)$ 和输出布尔函数  $C=(c_1c_2c_3)=f(S, b)$ 的布尔方程。这些方程是从图 4A 的状态转换表中得到的分析表达式。在这些布尔

表达式中，布尔变量 C 的非表示为 “'C”。因此，111 的非可以表示为 '(111)。需要说明的是，图 4B 中的布尔方程完全确定了与/或的组合逻辑(图 4C 中的 415)。

相关地，用于低占空比(1, 9)比率为 2/3 的 RLL 编码器的逻辑示于图 4C。在这个逻辑的实施例中，一个当前输入位对( $b_1b_2$ )通过锁存器 407 和 403 到达组合逻辑 415。同样，一个前瞻输入位对( $b_3b_4$ )通过锁存器 405 和 401 到达逻辑 415。被编码的输出 C 通过锁存器 417, 419 和 421 到达逻辑 415。下一个内部状态 N 通过分别对应输入锁存器 409, 411 和 413 的反馈路径 I, II, III 到达逻辑 415。

现在参照图 5A-5C，分别示出了根据本发明修改的一个用于(1, 9)比率为 2/3 的 RLL 译码器的状态转换表，涉及这个非受限输入返回到受限输出的布尔运算及该译码器的一个高级逻辑门的实施例。在图 5A 中，当提供用于前瞻几个 RLL 符号的时候，译码表是状态独立的。也就是，它在读出编码时不建立一个 FSM。这个 RLL 二进制编码的符号指定了从左到右的最初的三列。这些符号包括一个当前符号( $r_1r_2r_3$ )和两个前瞻符号( $r_4r_5r_6$ )和( $r_7r_8r_9$ )。第四列被指定为“译码输出”。为了译码，该 RLL 符号字符包括一个三位字符的滑动块，在这个滑动块中，根据图 5B 的布尔方程，将该当前符号与两个前瞻符号逻辑组合的结果作为当前符号的译码。

现在参照图 5C，每一个 RLL 编码的符号包括一个 3 位的字节，这 3 位分别到达锁存器 501, 503 和 507。通过或门 513，锁存器 519 和锁存器 521 处理这些位，形成三个中间的内部变量  $z_1$ ,  $z_2$  和  $z_3$ 。它们同时作为输入到达组合逻辑 525。进一步，当前的 RLL 编码的符号的第一位和第三位的值  $r_1$  和  $r_3$  作为同时存在的输入，通过各个锁存器 509, 517, 515 和 523 到达逻辑 525。

逻辑 525 由图 5B 中所示的用于译码输出的逻辑方程完全确定。当逻辑方程描述一个 6 代码位的输出( $u_1u_2u_3u_4u_5u_6$ )时，输出变量  $u_5$  被设置为 0。在所示的实施例中，附加逻辑用于将被译码输出 U 转换成 2 位字节的原始非受限二进制序列。因此，通过或门 529 主动与  $u_4$  和锁

寄存器 527 中的  $u_6$  的码型的组合以及或门 539 主动与锁存器 531 中的或门 529 的输出和逻辑 525 的输出  $u_2$  的组合, 来计算 2 位字节中的第一个字节。通过或门 535 主动与  $u_1$  和锁存器 533 中的  $u_3$  的一个码型的组合, 来计算 2 位字节中的第二个字节。2 位字节中的第一个字节放在锁存器 541 中, 而 2 位字节中的第二个字节放在锁存器 537 中。

现在参照图 6A-6C, 分别描述了根据本发明修改的用于一个(1, 13)比率为 2/3 的 RLL 编码器的状态转换表, 涉及非受限输入到受限输出的布尔运算, 以及编码器的一个高级逻辑门的实施例。仔细看一下图 6B, 图中所示是状态转换表, 该状态转换表的形式是由一个当前内部状态  $S$  和一个组合输入( $b_1b_2b_3b_4b_5b_6$ )做索引的内部状态和编码输出的有序对的一个阵列。对于一个(1, 13)编码器, 组合输入就是当前编码二进制输入位对和两个前瞻编码二进制输入位对。如果该编码器用于一个具有 00xxxx 形式的连续输入, 那么它将最终转换到状态 000, 并循环在该状态内产生一个占空比为 1/3 的具有 010010010010...形式的输出。和与输出 010101...相关输出的相比, 这仍远远少于 1/2。

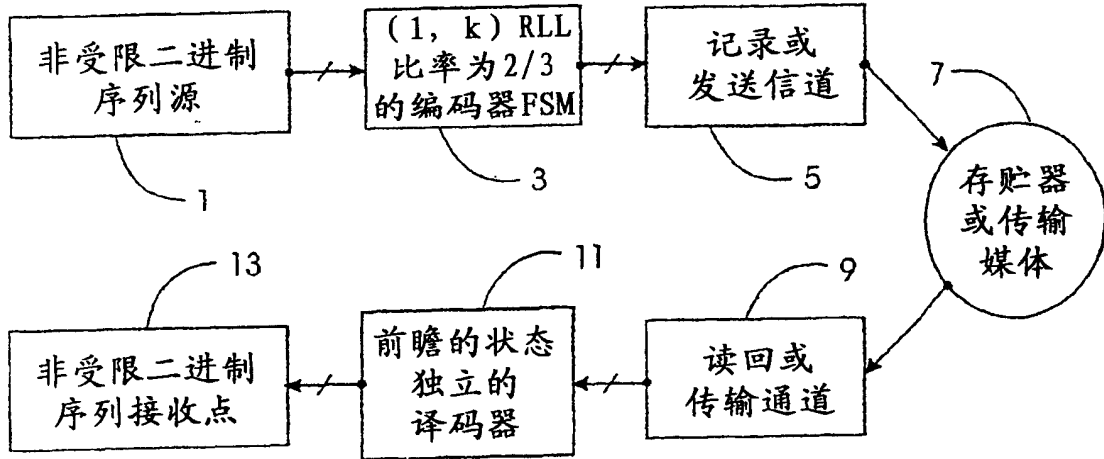
图 6A 所示的(1, 13)编码器的布尔方程与图 4B 所示的(1, 9)编码器的方程, 在总体结构上是非常相似的。也就是说, 他们源于状态转换表并被约束产生一个低占空比的输出。方程中通过在布尔变量的上方加一个水平横线, 来清楚地作为逻辑非的替代表示法。

现在参照图 6C, 图 6C 示出了一个(1, 13)编码器的逻辑实施例, 该实施例等同于图 4C 的(1, 9)编码器的逻辑实施例, 不同的是(1, 13)编码器是用独立接触形式。独立接触形式最早用于表示使用的中继电路网络, 例如, 在电话工业中, 通过和(1, 9)编码器相关描述实质相同的一种方法, 将三个输入位对的  $d_1, d_2$  送到缓冲器 L, 输入位  $b_1...b_6$  和内部状态一起送到组合逻辑。

现在参照图 7A-7C, 分别陈述了根据本发明修改的一个用于(1, 13)比率为 2/3 的 RLL 译码器的译码表, 涉及的受限输入到非受限输出布尔运算, 以及译码器的一个高级逻辑门的实施例。在图 7A 中, 陈述了包含一个当前 RLL 符号和三个前瞻 RLL 符号的状态独立的译码表,

---

用来执行将当前 RLL 符号映象到非受限二进制位对序列。相应地，图 7B 示出了定义译码器的布尔逻辑方程，图 7C 示出了该独立接触逻辑的实施例以及和前述的图 4C 建立的实施例相联系一种的运行方法。



在数据存储器或传输过程中具有低占空比的  
(1, 9或1, 13) RLL比率为2/3的编码器

图1

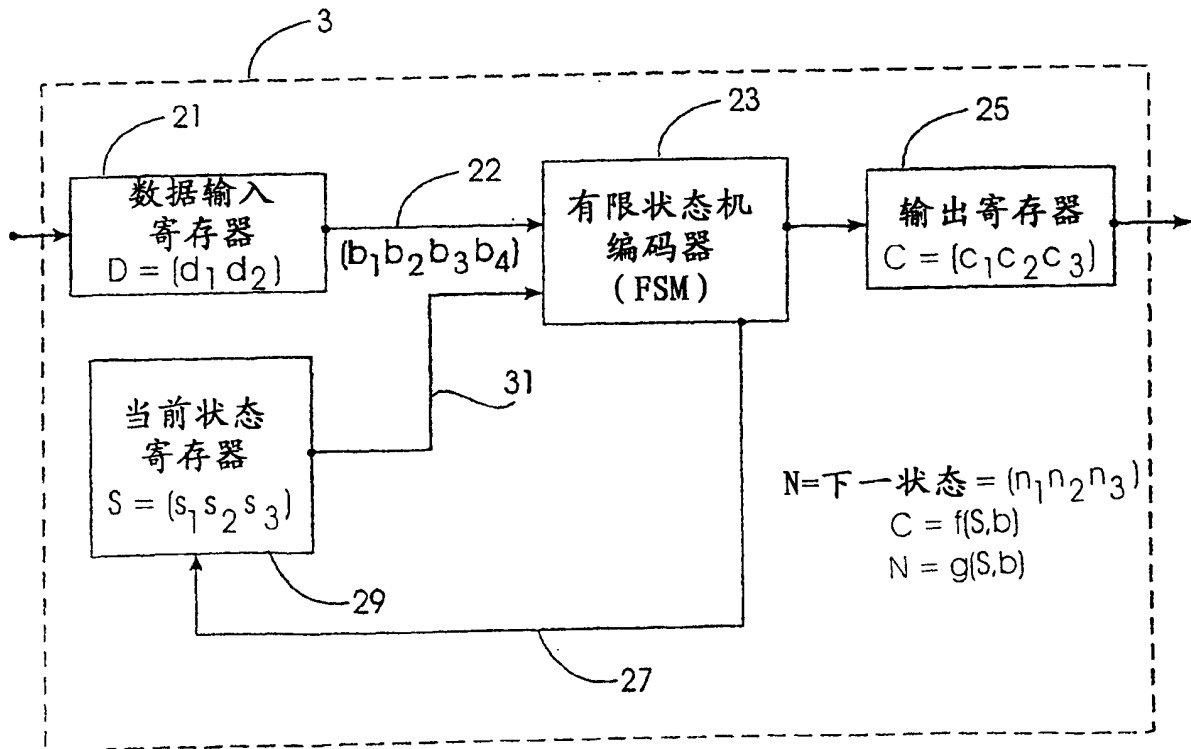


图2

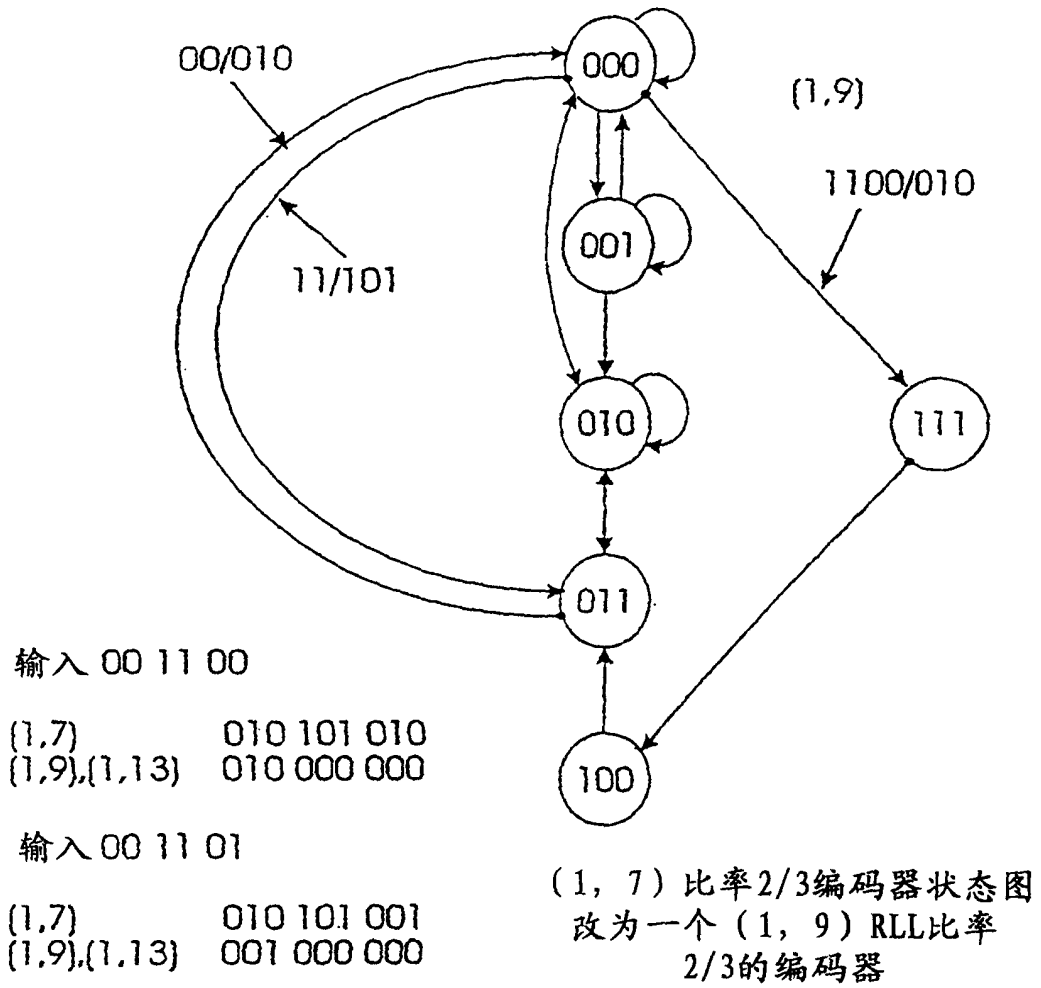


图 3

		下一状态/内部输出 $N = (n_1 n_2 n_3) / C = (c_1 c_2 c_3)$					
		内部输入 $b_1 b_2 b_3 b_4$ $x =$ 无关位值					
内部 状态 $S = s_1 s_2 s_3$	00xx	01xx	10xx	1100	1101	1110	1111
000	000/010	001/010	010/010	111/010	111/001	011/010	011/010
001	000/001	001/001	100/001	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	011/100	011/100	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000
111	100/000	100/000	100/000	100/000	100/000	100/000	100/000

用于低占空比 (1, 9) 比率 2/3 的 RLL 编码器的有限状态转换器

图 4A

下一状态布尔函数

$$n_1 = b_1 s_3 + b_1 b_2 b_3 s_1 s_2 + s_1 s_3$$

$$n_2 = b_1 s_3$$

$$n_3 = b_2 s_3 + s_1 b_1 b_2$$

输出布尔函数

$$c_1 = s_1 s_2$$

$$c_2 = s_1 s_2 c_3$$

$$c_3 = s_1 s_3 (b_1 + b_2) + s_1 s_2 s_3 b_1 b_2 b_3 b_4$$

定义 (1, 9) 比率 2/3 RLL 编码器的方程式

图 4B

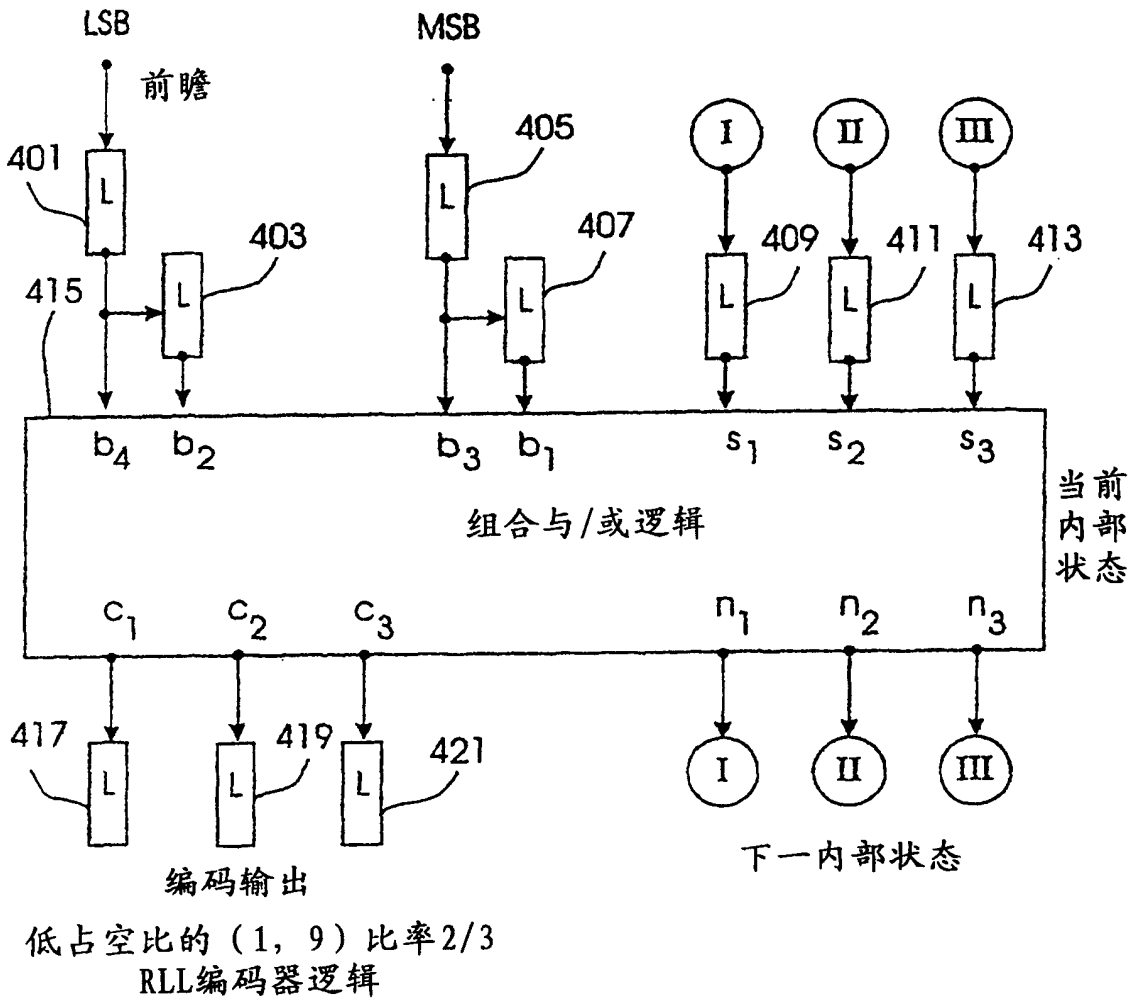


图 4C

$R = (r_1 r_2 r_3)$	$(r_4 r_5 r_6)$	$(r_7 r_8 r_9)$	译码输出
010	$\bar{0}00$	-----	00
001	$\bar{0}00$	-----	01
100	$\bar{0}00$	-----	10
101	$\bar{0}00$	-----	11
010	000	$\bar{0}00$	0111
001	000	$\bar{0}00$	0110
100	000	$\bar{0}00$	1111
101	000	$\bar{0}00$	1110
010	000	000	001100
001	000	000	001101

低占空比 (1, 9) 比率 2/3RLL 译码器的译码表

图 5A

内部变量

$$z_1 = (r_1 + r_2 + r_3); z_2 = (r_4 + r_5 + r_6); z_3 = (r_7 + r_8 + r_9)$$

译码输出的逻辑方程

$$U_1 = r_1$$

$$U_2 = (\bar{z}_2)r_3 + z_1z_2\bar{z}_3$$

$$U_3 = \bar{z}_1z_2$$

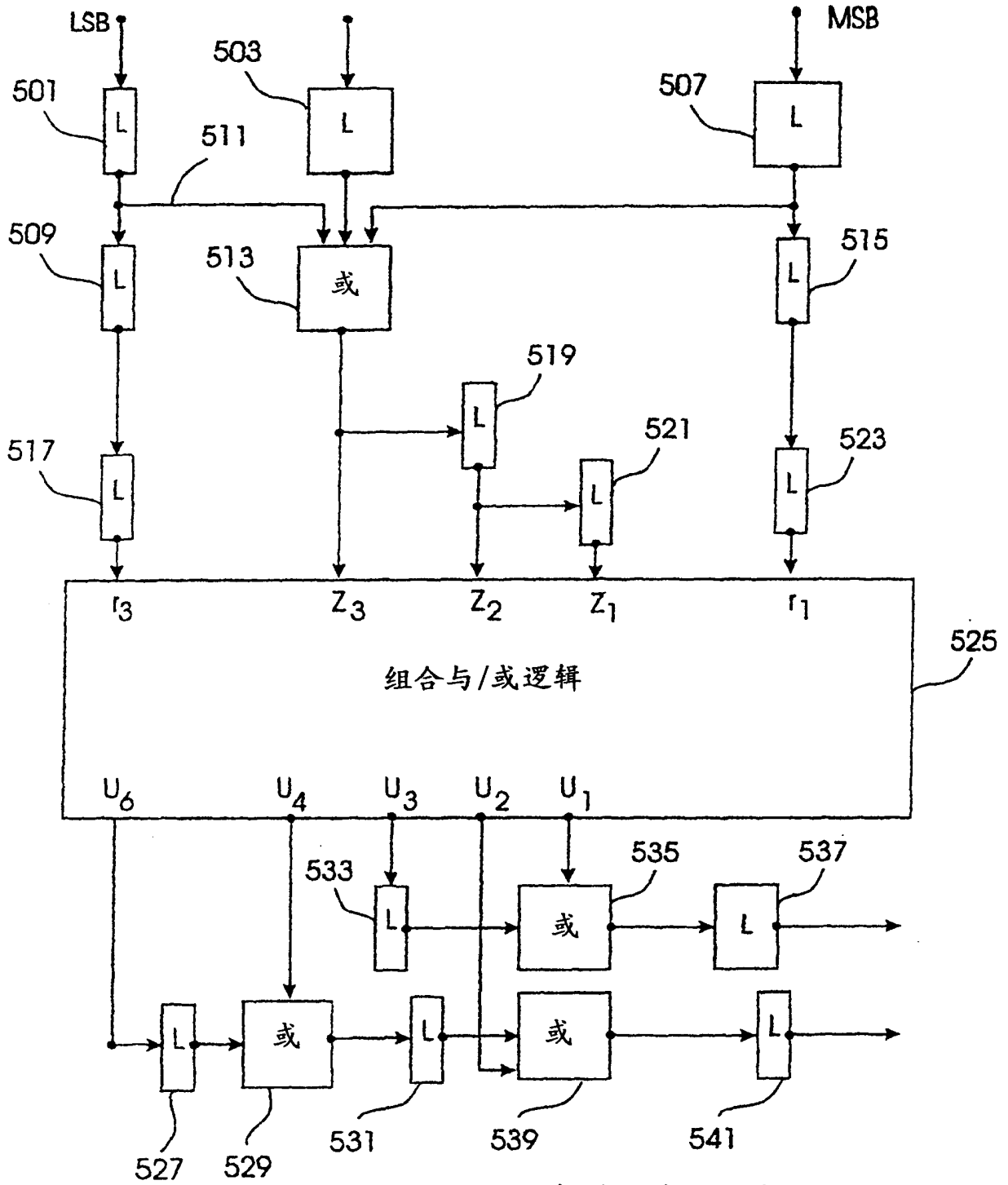
$$U_4 = z_2z_3 + \bar{z}_1z_2\bar{r}_3$$

$$U_5 = 0$$

$$U_6 = z_2z_3r_3$$

定义 (1, 9) 比率 2/3RLL 译码器的方程式

图 5B



低占空比 (1, 9) 比率 2/3 状态独立的前瞻译码器的逻辑

图 5C

确定下列矢量:

数据输入:  $D = (d_1, d_2)$

当前状态:  $S = (s_1, s_2, s_3)$

下一状态:  $N = (n_1, n_2, n_3)$

内部数据:  $B^1 = (B_1^1, B_2^1) = (b_1, b_2)$

$B^2 = (B_1^2, B_2^2) = (b_3, b_4)$

$B^3 = (B_1^3, B_2^3) = (b_5, b_6)$

内部代码字:  $C = (c_1, c_2, c_3)$

编码器输出:  $Y = (y_1, y_2, y_3)$

根据状态  $S$ ,  $B^1$ ,  $B^2$ , 和  $B^3$ , 通过下面布尔表达式来计算  $N$  和  $C$

$$n_1 = (s_1 s_3) + (s_3 b_1) + (\overline{s_1} b_1 b_2 \overline{b_3}) + (\overline{s_1} b_1 b_2 \overline{b_4} b_5 b_6),$$

$$n_2 = (\overline{s_3} b_1) + (s_1 s_2 b_1 \overline{b_2}),$$

$$n_3 = (\overline{s_3} b_2) + (\overline{s_1} \overline{b_1} b_2) + (s_1 s_2 b_1 \overline{b_2}),$$

$$c_1 = \overline{s_1} s_2,$$

$$c_2 = \overline{s_1} \overline{s_2} \overline{c_3},$$

$$c_3 = \overline{s_1} s_3 (\overline{b_1} + \overline{b_2}) + \overline{s_1} \overline{s_3} b_1 b_2 \overline{b_3} b_4.$$

矢量  $B^1$ ,  $B^2$ ,  $B^3$ ,  $S$  和  $Y$  是锁存器输出, 它更新如下

$$B^1 \leftarrow B^2 \leftarrow B^3 \leftarrow D,$$

$$S \leftarrow N, \text{ AND } Y \leftarrow C.$$

确定低占空比 (1, 13) 比率 2/3 编码器的方程式

图 6A

下一状态/内部输出: N = (n1 n2 n3)/C = (c1 c2 c3)							
内部输入 (b1 b2 b3 b4 b5 b6) X = 无关位值							
内部状态 S = (S1S2S3)	00XXX	01XXX	10XXX	1100XX	1101XX	1110'(11)	1111XX
000	000/010	001/010	010/010	111/010	111/001	011/010	011/010
001	000/001	001/001	100/001	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	111/100	111/101	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000
111	100/000	100/000	111/000	100/000	100/000	100/000	100/000

用于低占空比 (1, 13) 比率 2/3 RLL 编码器的有限状态转换表

图 6B

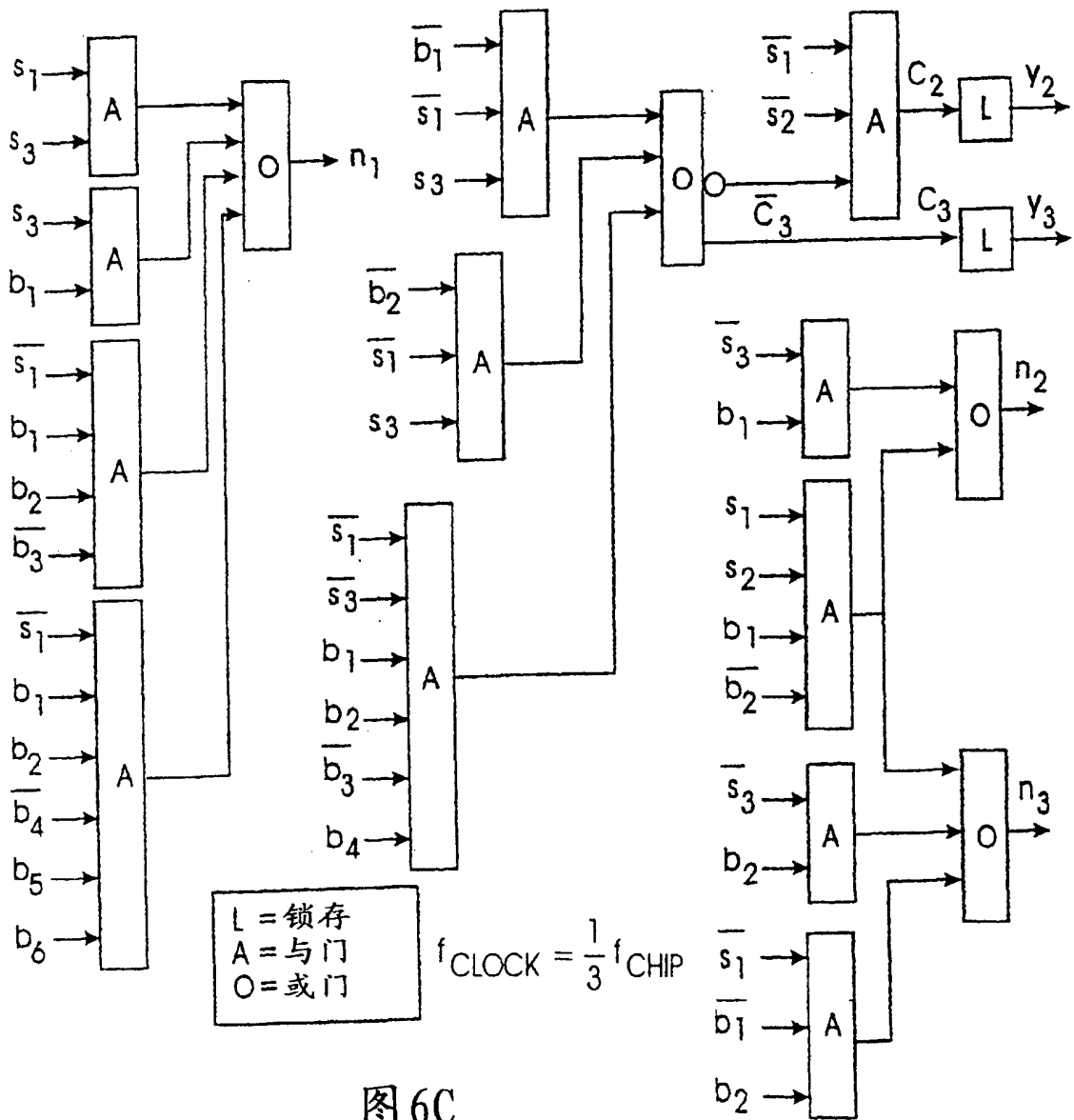
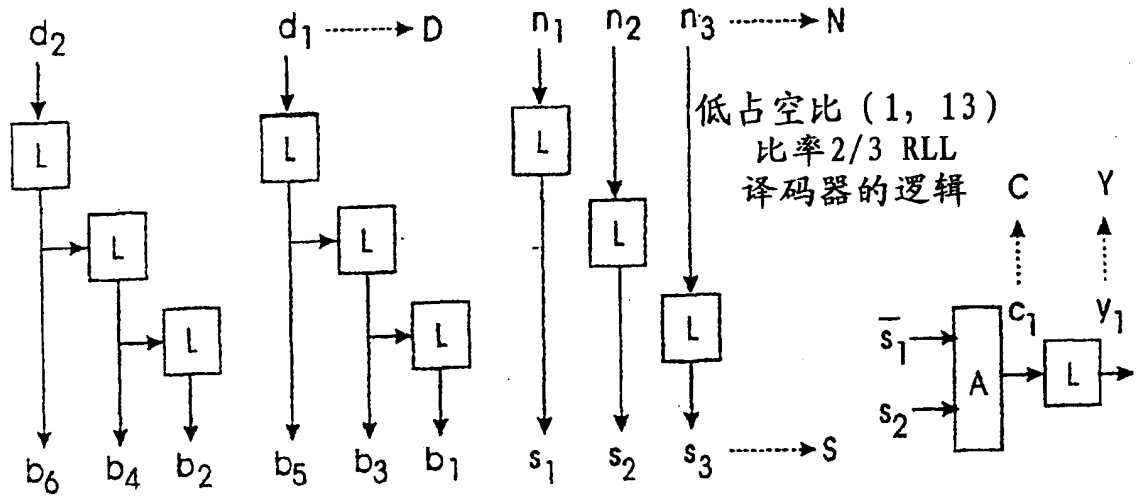


图 6C

$R = (r_1 r_2 r_3)$	$(r_4 r_5 r_6)$	$(r_7 r_8 r_9)$	$(r_{10} r_{11} r_{12})$	译码输出
010	$\bar{\phantom{0}}(000)$			00
001	$\bar{\phantom{0}}(000)$			01
100	$\bar{\phantom{0}}(000)$			10
101	$\bar{\phantom{0}}(000)$			11
010	000	$\bar{\phantom{0}}(000)$		0111
001	000	$\bar{\phantom{0}}(000)$		0110
100	000	$\bar{\phantom{0}}(000)$		1111
101	000	$\bar{\phantom{0}}(000)$		1110
010	000	000	$\bar{\phantom{0}}(000)$	001100
001	000	000	$\bar{\phantom{0}}(000)$	001101
100	000	000	$\bar{\phantom{0}}(000)$	101100
101	000	000	$\bar{\phantom{0}}(000)$	101101
010	000	000	000	00111011
100	000	000	000	10111011

低占空比 (1, 13) 比率2/3 RLL译码器的译码表

图7A

确定矢量:

接收到的代码字:  $R = (r_1, r_2, r_3),$

内部代码字:  $Y^4 = (y_{10}, y_{11}, y_{12}),$

$Y^3 = (y_7, y_8, y_9),$

$Y^2 = (y_4, y_5, y_6),$

$Y^1 = (y_1, y_2, y_3),$

内部变量:  $Z_B = y_4 + y_5 + y_6,$

$Z_C = y_7 + y_8 + y_9,$

$Z_D = y_{10} + y_{11} + y_{12},$

$X^1 = (x_1^1, x_2^1) = (x_1, x_2),$

$X^2 = (x_1^2, x_2^2) = (x_3, x_4),$

$X^3 = (x_1^3, x_2^3) = (x_5, x_6),$  确定 (1, 13) 比率 2/3

$U = (u_1, u_2),$

RLL译码器的方程式

$V = (v_1, v_2),$

$W = (w_1, w_2).$

变量  $X^1, X^2$  和  $X^3$  由下面布尔表达式计算出来:

$$x_1 = v_1,$$

$$x_2 = (y_6 \overline{Z_C}) + (\overline{Z_B} Z_C \overline{Z_D}) + v_2,$$

$$x_3 = (Z_B Z_C Z_D) + (\overline{Z_B} Z_C) + w_1 + w_2,$$

$$x_4 = (Z_B Z_C \overline{Z_D} y_3) + (\overline{Z_B} Z_C [Z_D + \overline{y_6}]) + w_2,$$

$$x_5 = y_{10},$$

$$x_6 = Z_B Z_C Z_D,$$

矢量  $Y^1, Y^2, Y^3, Y^4, U, V$  和  $W$  是锁存器输出;

它们更新如下:

$$Y^1 \leftarrow Y^2 \leftarrow Y^3 \leftarrow Y^4 \leftarrow R,$$

$$W \leftarrow X^3, V \leftarrow X^2, U \leftarrow X^1,$$

在此  $U$  代表译码的数据位

图 7B

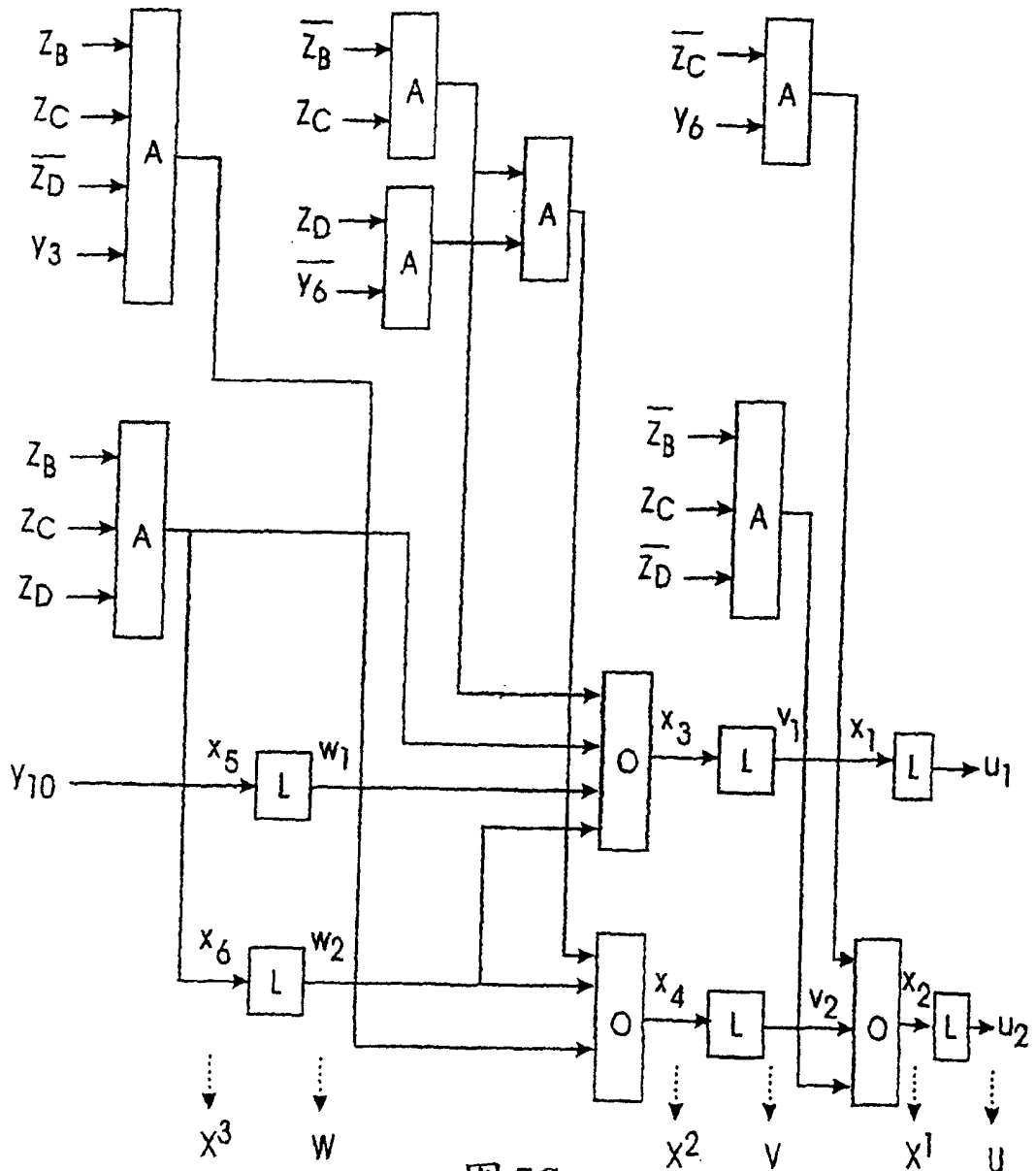
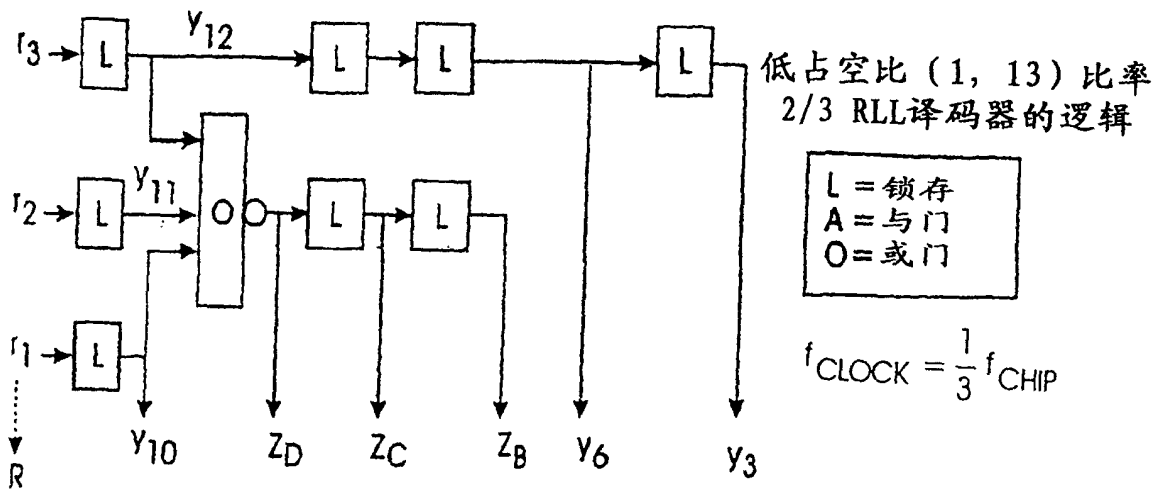


图 7C