



[12] 发明专利申请公开说明书

[21] 申请号 97121148.5

[43]公开日 1998 年 7 月 1 日

[11] 公开号 CN 1186385A

[22]申请日 97.10.22

[30]优先权

[32]96.10.30[33]GB[31]9622540.4

[71]申请人 迪维安公司

地址 美国加利福尼亚州

[72]发明人 理查德·J·托马斯

[74]专利代理机构 柳沈知识产权律师事务所

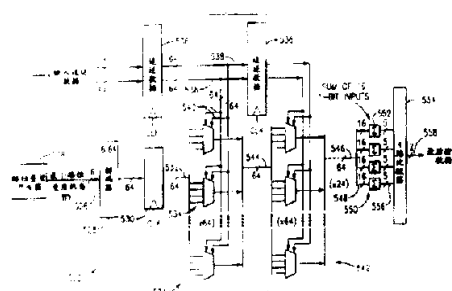
代理人 马 莹

权利要求书 2 页 说明书 14.0 页 附图页数 27 页

[54]发明名称 用于维特比解码器的追逆

[57]摘要

在用于卷积编码数据流的 M-阶维特比解码器的追逆单元中，各追逆级有一组表示 K 个可能候选状态的 K 个输入导线。K 个多路复用器的组合根据 M 比特追逆数据选择 2^M 个输入导线之一。K 个多路复用器输出馈送到后续追逆级。M 组 K 条导线承载该追逆数据，每条导线与每个多路复用器的一条选择线连接。在追逆单元的输出端，标识电路标识具有最大数量候选状态的 K 个可能性的一个子组，从而消除了在每个追逆级之间需重新定时的需要。



权 利 要 求 书

1. 一种用于卷积编码数据流的 M - 阶维特比解码器, 该解码器具有一个用于 K 个状态的路径量度计算器, 和多个追逆级, 所述追逆级在一个操作
5 周期中从所述路径量度计算器接收一最小路径量度状态标识和追逆数据, 其中改进包括:

多个追逆级, 每一追逆级包括:

第一组 K 条导线, 其上具有表示一级操作的候选状态, 所述第一组定义了多个输入子组;

- 10 一个 K 个选择器的组合, 每个选择器具有被映射到一所述输入子组的 N 个输入, 每个选择器还具有用于接受 M 比特追逆数据的多条选择线; 和所述组合具有连接到后续追逆级的 K 个输出端, 其中 $N=2^M$; 和

用于承载追逆数据的 M 组 K 条导线, 其中所述 M 组中每组的一预定导线被操作地映射到每个所述选择器的一条所述选择线上。

- 15 2. 根据权利要求 1 的维特比解码器, 其中所述 K 个输出定义了多个输出子组, 所述解码器还包括一个标识电路, 用于确定所述输出子组中一个所选子组, 该所选子组具有最大数量的所述候选状态。

3. 根据权利要求 2 的维特比解码器, 其中所述标识电路包括:

- 20 多个加法器, 每个加法器与所述 K 个输出的一个所述输出子组连接, 用于把这些输出相加;

一个与所述加法器耦合连接的比较器电路, 用于识别与此相关联的一个输出子组。

4. 根据权利要求 1 的维特比解码器, 进一步包括多个重新定时寄存器, 其中所述多个追逆级在数量上多于所述重新定时寄存器。

- 25 5. 一种用于卷积编码数据流的 M - 阶维特比解码器, 该解码器具有一个用于 K 个状态的路径量度计算器, 和多个追逆级, 所述追逆级在一个操作周期中从所述路径量度计算器接收一最小路径量度状态标识和 M 比特追逆数据, 其中改进包括:

多个追逆级, 每一追逆级包括:

- 30 第一组 K 条导线, 其上具有表示一级操作的候选状态, 所述第一组定义了多个输入子组, 每个所述输入子组具有 N 个成分, 其中 $N=2^M$;

一个 K 个多路复用器的组合, 每个多路复用器具有被映射到一所述输入子组的 N 个输入, 每个多路复用器还具有 M 条选择线, 和一个与一后续追逆级连接的输出; 其中在所述追逆级的最后一级中, 所述 K 个多路复用器的组合的所述输出定义了多个输出子组, 每个所述输出子组具有 K/N 个成分;

- 5 用于承载追逆数据的 M 组 K 条导线, 其中所述 M 组中每组的一预定导线被操作地映射到每个所述多路复用器的一条所述选择线上;

多个加法器, 每个加法器与一所述输出子组连接, 用于把这些输出相加;

- 10 一个与所述加法器耦合连接的比较器电路, 用于识别与此相关联的一个输出子组.

6. 根据权利要求 5 的维特比解码器, 进一步包括用于所述 M 组 K 条导线的多个重新定时寄存器, 其中所述多个追逆级在数量上多于所述重新定时寄存器。

用于维特比解码

器的追逆

5

本发明涉及一种卷积编码数据流的解码器。确切地说，本发明涉及一种用于维特比(Viterbi)解码器的改进的追逆装置。

维特比解码器在现代电信中是很重要的，数字信号在维特比解码器中被卷积解码。在一种现代应用中，要求卷积编码的正交编码频分复用(“COFDM”)已在欧洲电信标准 DRAFT pr ETS 300 744(1996年5月)中提出，其规定了帧的结构，信道编码，和用于数字陆上电视的调制方式。该标准被设计以适应处在现存用于模拟传输的频谱位置内的数字陆上电视，还提供了充足的保护以免受公共信道的高电平干扰和相邻信道干扰。该标准规定了一个适应性强的保护间隔，以便系统能够支撑各种不同的网络结构，同时维持高频谱效率和充足的保护以免受来自现存 PAL/SECAM 业务的公共信道的高电平干扰和相邻信道干扰。还定义了两种工作模式：“2K 模式”，适用于单发射机工作和具有有限发射机距离的小的单频率网络，和“8K 模式”。后者既可用于单发射机工作也可用于大的单频率网络。如同是不同的内部编码比率，正交幅度调制(“QAM”)的各种电平都得到支撑，以便使比特率与耐久性相平衡。该系统意图是要适应根据“运动图像专家组(MPEG)”的传输层，和直接与 MPEG - 2 编码 TV 信号兼容(ISO/IEC 13818)。

在著名的欧洲电信标准数据载波中，COFDM 帧可以是正交相移键控(“QPSK”)，16 - QAM，64 - QAM，非均匀 16 - QAM，也可以使用 Gray 映射的非均匀 64 - QAM。

25 在下面的讨论中可以参考图 1，图 1 示出一个根据一阶维特比编码过程以 1/2 编码比率、按照具有有限长度 $K=3$ 且生成多项式为 $G(x)$ 的卷积编码的简化传输格形图 167，其中

$$G(x)=(x^2 + x + 1, x^2 + 1) \quad (1)$$

比率 1/2 表示，对于每一比特输入编码器产生两个比特。有限长度 K 是能被用于产生一输出的信号的最大数。由于利用了诸如图 167 的传输格形图和一输入数据序列，因此有可能产生一个跟随在一状态 S 序列之后的输出流。在

图 167 中, 一特定状态 S_t 可用两个比特表示。例如, 状态 S_t 可以表现为值 2(二进制 10), 用参考数字 169 表示。在图 167 的表示中, 在状态 S_{t+1} 中, 状态 S_t 被移动一个位置, 且一输入数据比特占据最右(最低有效位)位置。这样, 状态值 169 可合法地转移到状态 S_{t+1} 中的值 171 和 173。对于这两次转移, 卷积编码器将分别产生值 175 和 177, 其被更一般地表示为 $X_t Y_t$ 。可为该编码器计算所有可能的状态转移, 即给定 S_t 和数据比特 d_t , 则下一状态 S_{t+1} , X_t 和 Y_t 能被估计得出。

路径量度(路径 metric)是所述状态在那一时刻处在原始编码器状态序列上的最大似然的度量。路径量度越小, 该状态的可能性越大, 反之亦然。分支量度是与每一分支相联取决于输入的概率值的度量。该分支量度被作为汉明加权值, 它是在每一如图 1 所示的每一次转移中沿每一分支的一被接收码元 XY_{rx} 和一被期望码元 XY 之间的不同比特数。追逆是通过格形追逆以找到产生具有最小路径量度的状态的初始状态的方法。

在本优选实施例中, 使用了一种 2 阶解码过程, 其对应着通过格形移动一次两步。这加倍了计算每一步的时间, 且每一次追逆产生两个比特而不是一个。然而, 在每一状态要求计算的次数也加倍, 因为在此时具有四种要被计算的可能路径。被称作幸存路径的路径是具有最小路径量度的路径, 因而是最可能的路径。

收缩(puncturing)具有产生较高数据传输速率的效果, 因为这种编码是非常有效的。在列表 2 中, 卷积编码器(未示出)对数据进行编码以产生码元 X_t 和 Y_t , 随后根据收缩矩阵

$X: 10$

$Y: 11$

使 X_t 和 Y_t 收缩以产生 X'_t 和 Y'_t , 随后它们被重新定时, 按正交相移键控调制作为 I、Q 发送。当通过收缩数据进行解码时, 被删除比特不对分支量度做出贡献。

表 1

data	d_0	d_1	d_2	d_3	d_4
xy	x_0y_0	x_1y_1	x_2y_2	x_3y_3	x_4y_4
$x'y'$	x_0y_0	y_1	x_2y_2	y_3	x_4y_4
IQ	x_0y_0	y_1x_2	y_2y_3	x_4y_4	

在上述所给样例中，利用了汉明加权对分支量度进行计算。如果我们接收表示该为 1 或 0 的信号的相对似然的每个 X_{rx} 和 Y_{rx} 的一个多比特表示(不是接收 1 或 0)，则有效的改进产生。这样，在 16 - 级(4 - 位)软解码中，一个 1 由 15(二进制 1111)表示。

在 16 级解码中，例如，如果接收到 $XY_{rx}=(3, 14)$ ，则分支量度可如表 2 所示计算。当计算新的路径量度时，利用这些被计算的软分支量度计算各路径量度，从而对解码器性能做出有效的改进。在本优选实施例，使用了 8 级(3 位)软解码。正如下面所详细描述，使用了一个 systolic 矩阵实现追逆。

表 2

期望的 xy_{rx}	分 支	计 算	结 果
00	0	$ 0 - 3 + 0 - 14 $	17
01	1	$ 0 - 3 + 15 - 14 $	4
10	2	$ 15 - 3 + 0 - 14 $	26
11	3	$ 15 - 3 + 15 - 14 $	13

在该优选实施例中，利用一约束长度 $K=7$ 对数据进行卷积解码，约束长度 $K=7$ 对应着一个具有 64 个状态的格形。图 2 中示出了用于这种情形的一个 2 阶转移格形图的一部分表示。

在著名的欧洲电信标准中有一种外层 Reed - Solomon 码，和一种以一具有 64 状态，比率 1/2 的母卷积码为基础的内层收缩卷积码，且对于 X 输出有生成多项式 $G_1=171_{OCT}$ 和对于 Y 输出有生成多项式 $G_2=133_{OCT}$ 。在一种操作模式中，也可允许使用其他的如 2/3, 3/4, 5/6 和 7/8 的收缩比率。

在维特比解码过程期间，追逆需要一个处理时间的基本量和硬件资源。维特比解码器中的追逆模块也代表着一个重要的生产成本。

因而，本发明的一个基本目的是提供一种在对卷积编码数据解码的系统

中使用的改进型维特比解码器。

本发明的另一个目的是提供一种维特比解码器中的改进型追逆装置，利用现代硬件设计方法对其进行生产可做到很经济，且使硬件资源最小化。

5 本发明的再一个目的是提供一种用于维特比解码器的追逆装置，其可被方便地应用在各种电信和其他技术中的应用中。

10 本发明的这些和其他目的是通过一个用于卷积编码数据流的 M - 阶维特比解码器实现。该解码器具有一个用于 K 个可能状态的路径量度计算器，和多个追逆级，在一个操作周期中其从路径量度计算器接收一最小路径量度和追逆数据。在每个追逆级中，一组 K 个输入导线表示一级操作的候选状态，和定义了多个输入子组。在一个 K 个选择器的组合中，该选择器可以是多路复用器，每个选择器具有被映射到一输入子组的 N 个输入，和具有用于接受 M 比特追逆数据的多条选择线，其中 $N=2^M$ 。 K 个选择器的组合具有连接到后续追逆级的 K 个输出端。 M 组 K 条导线承载追逆数据，该 M 组中每组的一预定导线被操作地映射到每个选择器的一条选择线上。

15 K 个输出定义了多个输出子组。该解码器具有一个识别电路，用于确定哪个输出子组具有保持在此的最大候选状态数。

根据本发明的一个方面，该识别电路包括多个加法器，其中每个都与输出子组之一连接用于使该输出相加，和一个与所述加法器耦合的比较器电路。

20 根据本发明的另一个方面，具有多个重新定时寄存器，其中追逆级的数量大于重新定时寄存器的数量。

为了更好地理解本发明的这些和其他目的，请参考本发明通过举例所做的详细描述并参考附图，其中：

图 1 说明根据一阶维特比解码过程的简化转移格形图；

25 图 2 是根据在本发明的优选实施例中执行的维特比解码过程的转移格形图的局部示图；

图 3A - 3B(总地称为图 3)是一个维特比解码器的框图；

图 4A - 4B(总地称为图 4)是图 3 所示维特比解码器中一控制单元的电气示意图；

30 图 5， 6 和 7 说明了图 4 所示控制单元中的解码逻辑；

图 8 是图 3 中所示维特比解码器的分支量度产生框的电气示意图；

图 9A - 9F(总地称为图 9)是更详细说明图 8 的电路的一部分的电气示意图;

图 10 是在图 9 的电路中一计算单元的详细电气示意图;

图 11 是使用在图 10 中所示电路中的一个逻辑网络的电气示意图;

5 图 12 是图 9 的电路中一求和单元的电气示意图;

图 13A - 13B(总地称为图 13)是图 3 中所示维特比解码器的路径量度产生框中的一相加 - 比较 - 选择单元的电气示意图;

图 14 是说明一路径量度的计算的示图;

10 图 15 是说明图 13 中所示维特比解码器的路径量度产生框中的一相加 - 比较 - 选择单元的部分框图;

图 16 是图 13 中所示维特比解码器的路径量度产生框中的一相加 - 比较 - 选择单元的一部分的电气示意图;

图 17 是说明图 3 中所示维特比解码器的控制框操作顺序的流程图;

图 18A - 18B(总地称为图 18)是图 17 中描述的控制框的电气示意图;

15 图 19 是图 3 中所示维特比解码器中的追逆单元的追逆纵列的电气示意图;

图 20 是维特比解码器的一个追逆单元的电气示意图; 和

图 21 是根据本发明的一个优选实施例的维特比解码器的一个追逆单元的电气示意图。

20 参考图 3, 一个维特比解码器具有一个旋转调整框 179, 其在线路 48, 50 上接收同相和正交数据。输入数据的有效性由线路 52 上一信号的状态指使。在前级中, 可以按任何 8 个载波旋转相位锁定该信号星座, 考虑到接收到的信号谱有可能被颠倒, 在这种意义下, I, Q 码元被接收, 成为 Q, I 码元。然而, 正如下面所讨论, 这种情况并不在旋转调整框 179 中。当前的
25 旋转相位被设置在总线 181 上。

对于所有可能的分支(在本优选实施例中为 256, 利用了约束长度 $K=7$, 和 2 - 阶编码), 在分支量度产生框 186 中在每个操作周期计算各分支量度, 该分支量度产生框 186 由相位和收缩控制框 188 控制。总线 185 上的当前收缩比率和总线 187 上的当前收缩相位从一个更高的控制框, 即维特比控制框
30 195 被输入到相位和收缩控制框 188。分支量度产生框 186 中的各状态被映射和选择, 以便根据线路 197, 199, 201, 和 203 的状态进行收缩和相位调整。

图 4 中更详细地示出了相位和收缩控制框 188。通过组合逻辑网络 205 由总线 185 上的收缩比率导出最大相位数，并放置到总线 207 上。相位计算在部分 209 由一个 3 - 位加法器(总的用 211 表示)，及跟在其后的一个以 213 表示的 4 - 位减法器完成。对相位计算值，最大相位数求模进行确定，并将其提交到相位计算部分 215，在部分 215 中，以与部分 209 相同方式该当前收缩相位与所述相位计算值相加。当前相位、最大相位数求模呈现在总线 217 上。因为在被收缩数据流的数据传输速率与系统处理速率之间存在着差，所以必须根据收缩相位使维特比解码器启动和停止。一个小逻辑网络 221 生成一个全局启动信号 219。

10 相位计算部分 215 的输出还在框 233 中被使用，以对总线 187 和 217 上的相位和比率信息进行解码，产生总线 197, 199, 201 和 203 上的各个信号。这些总线与分支量度产生框 186(图 3)通信连接。在图 5, 6 和 7 中示出了总线 197 和 199 的解码逻辑，其中各个比特位置在线路 225, 227, 228, 229, 231, 233 和 236 上标出。总线 201 和 203(图 3)上的信号是分别对总线 197 和 199 上的信号的补码。正如将在下面看到的，在分支产生期间，总线 199 在 I, Q 输入中做出一个选择，总线 203 指示收缩已在哪里发生，以便在该收缩位置的数据不再对路径量度计算做出贡献。

图 8 更详细地示出了分支量度产生框 186(图 3)。I, Q 数据对在线路 238, 240 上被分别接收，并根据选择器线路 225, 227, 228, 229 和在线路 252 和 254 上的在先 I, Q 数据在四个组合逻辑单元中被处理。所述选择器线路 225, 227, 228, 229 是线路 197(被表示为线路 244, 246, 248, 250)的补。根据该数据，对码元 XY 对重新组构并在线路 258, 260, 262, 264 上送到框 256，以便在线路 266 上产生所有 16 个可能的分支量度。从两个延迟触发器 268, 270 中获得所述在先 I, Q 数据。

25 图 9 中更详细地示出了框 256，其包括 16 个一样的计算单元，其中每一个对应着 16 个可能的分支之一。图 10 中还进一步描述了一个代表性的计算单元 272。每个计算单元 272 包括四个模块 274，其中所期望数据被实接线。各模块 274 以表 3 所示的方式对在输入数据与该特定分支的所期望数据之间的各绝对差求和，并根据线路 201 和 203 的状态迫使与被收缩位置对应的数据到零。如图 11 所示，模块 274 包括一个取样逻辑网络，其通过反转交替位确定绝对差。于是，在线路 278, 280, 282, 283 上获得的四个差在求和单

元 284 中被求和, 其在图 12 中被详细示出, 并在线路 266 输出作为分支量度。

现在参考图 3 和 13, 利用从分支量度产生框 186 中线路 288 上获得的
 预计算分支量度在路径量度产生框 189 中计算路径量度。分支量度的预计算
 大大地简化了路径量度的计算。路径量度产生框 189 在一个时钟周期能够处
 5 理 2 码元。通过合适的硬件设计选择, 分支量度产生框 186 和路径量度产生
 框 189 能被随意地综合以利用一个 M - 级维特比解码器在一个单一时钟周期
 中处理 M 码元。

路径量度产生框 189 的构成在图 14 中以举例方式初步示出, 其中示出
 了对于状态 0 的一个路径量度的计算。图 2 中所示格形图的所有扩展表明在
 10 S_{t+1} , 状态 0 能够合法地接收来自状态 0, 16, 32 和 48 的转移(分别用 290, 292,
 294, 和 296 标识)。这些转移在表 4 中被分别表示为路径 0 - 3, 对应着它
 们的状态数的两个最有效位。表 4 给出了在图 2 的转移格形图中每个合法转
 移的分支量度。从表 4 可以看到, 对于状态 0, 路径 0 具有期望分支数据
 0; 路径 1 期望 14; 路径 2 期望 3; 和路径 3 期望 13。利用接收到的数据,
 15 为每一所期望分支数据计算分支量度。为了为状态 0 的下一路径量度计算四
 个可能的候选分支数据, 利用加法器 298 将在先的状态 0 的路径量度加到具
 有所期望数据 0 的分支量度上。以便将候选 0 给到线路 300 上, 等等其他三
 个路径。此后, 在这四个候选者在框 302 中比较之后, 状态 0 的下一个路径
 量度是最小的候选值, 并在线路 304 上被输出。其他候选数据被放弃, 因为
 20 它们是非最佳路径。

被传给追逆的两个数据比特指示哪个路径被选作具有最小的路径量
 度, 即路径 0, 1, 2 或 3, 如所需按时间追逆。

表 3

状态 0: 路径 0=0 1=14 2=3 3=13	状态 32: 路径 0=8 1=6 2=11 3=5
状态 1: 路径 0=12 1=2 2=15 3=11	状态 33: 路径 0=4 1=10 2=7 3=9
状态 2: 路径 0=7 1=9 2=4 3=10	状态 34: 路径 0=15 1=1 2=12 3=2
状态 3: 路径 0=11 1=5 2=8 3=6	状态 35: 路径 0=3 1=13 2=0 3=14
状态 4: 路径 0=13 1=3 2=14 3=0	状态 36: 路径 0=5 1=11 2=6 3=8
状态 5: 路径 0=1 1=15 2=2 3=12	状态 37: 路径 0=9 1=7 2=10 3=4
状态 6: 路径 0=10 1=4 2=9 3=7	状态 38: 路径 0=2 1=12 2=1 3=15
状态 7: 路径 0=6 1=8 2=5 3=11	状态 39: 路径 0=14 1=0 2=13 3=3

状态 8: 路径 0=15 1=1 2=12 3=2	状态 40: 路径 0=7 1=9 2=4 3=10
状态 9: 路径 0=3 1=13 2=0 3=14	状态 41: 路径 0=11 1=5 2=8 3=6
状态 10: 路径 0=8 1=6 2=11 3=5	状态 42: 路径 0=0 1=14 2=3 3=13
状态 11: 路径 0=4 1=10 2=7 3=9	状态 43: 路径 0=12 1=2 2=15 3=1
状态 12: 路径 0=2 1=12 2=1 3=15	状态 44: 路径 0=10 1=4 2=9 3=7
状态 13: 路径 0=14 1=0 2=13 3=3	状态 45: 路径 0=6 1=8 2=5 3=11
状态 14: 路径 0=5 1=11 2=6 3=8	状态 46: 路径 0=13 1=3 2=14 3=0
状态 15: 路径 0=9 1=7 2=10 3=4	状态 47: 路径 0=1 1=15 2=2 3=12
状态 16: 路径 0=3 1=13 2=0 3=14	状态 48: 路径 0=11 1=5 2=8 3=6
状态 17: 路径 0=15 1=1 2=12 3=2	状态 49: 路径 0=7 1=9 2=4 3=10
状态 18: 路径 0=4 1=10 2=7 3=9	状态 50: 路径 0=12 1=2 2=15 3=1
状态 19: 路径 0=8 1=6 2=11 3=5	状态 51: 路径 0=0 1=14 2=3 3=13
状态 20: 路径 0=14 1=0 2=13 3=3	状态 52: 路径 0=6 1=8 2=5 3=11
状态 21: 路径 0=2 1=12 2=1 3=15	状态 53: 路径 0=10 1=4 2=9 3=7
状态 22: 路径 0=9 1=7 2=10 3=4	状态 54: 路径 0=1 1=15 2=2 3=12
状态 23: 路径 0=5 1=11 2=6 3=8	状态 55: 路径 0=13 1=3 2=14 3=0
状态 24: 路径 0=12 1=2 2=15 3=1	状态 56: 路径 0=4 1=10 2=7 3=9
状态 25: 路径 0=0 1=14 2=3 3=13	状态 57: 路径 0=8 1=6 2=11 3=5
状态 26: 路径 0=11 1=5 2=8 3=6	状态 58: 路径 0=3 1=13 2=0 3=14
状态 27: 路径 0=7 1=9 2=4 3=10	状态 59: 路径 0=15 1=1 2=12 3=2
状态 28: 路径 0=1 1=15 2=2 3=12	状态 60: 路径 0=9 1=7 2=10 3=4
状态 29: 路径 0=13 1=3 2=14 3=0	状态 61: 路径 0=5 1=11 2=6 3=8
状态 30: 路径 0=6 1=8 2=5 3=11	状态 62: 路径 0=14 1=0 2=13 3=3
状态 31: 路径 0=10 1=4 2=9 3=7	状态 63: 路径 0=2 1=12 2=1 3=15

5 路径量度产生框 189 包括 64 个相加 - 比较 - 选择框 306，其中的一个在图 15 中的局部示图中被完整地示出。每个相加 - 比较 - 选择框 306 都产生一个路径量度。在实际的布局中，以两组 32 个按排该 64 个相加 - 比较 - 选择框 306 是很方便的。这是一种比较紧凑的设计，它使长度最小化，因而使路径量度路由交叉的驱动要求最小化。在路径量度产生框 189 中确定 64 个路径量度中最小的。每个相加 - 比较 - 选择框 306 在每一 6 - 位总线 308，310，

312, 314 上接收一在先状态的四个路径量度, 对应着例如图 2 所示的转移。
在 5 - 位总线 316, 318, 320 和 322 上接收该相应的分支量度。

图 13 更详细地示出了相加 - 比较 - 选择框 306 的安排, 其中对输入到此的 4 个候选路径量度的最小路径量度进行确定。在加法器 324 中通过将在总线 326 和 328 上分别输入的一路径量度和一分支量度相加获得所述候选路径量度。然后在比较模块 330 中确定最小路径量度。在图 16 中更详细示出了比较模块 330, 其中对四个值进行比较以找出最小值。在图 16 左侧的 6 个单元 332 中, 做出各种可能的比较并将其在线路 334, 336 上输出。对于每个相应的单元 332, 线路 336 上的输出只是线路 334 上输出的反相。随后, 该结果在逻辑网络 338 中被解码并被放置到一个四位选择总线 340 上。这种安排的一个优点是可将相加, 比较, 和再度量构成流水线以节省时间。

再参考图 3 和 13, 在每个相加 - 比较 - 选择框 306 的每个数据操作周期, 两比特信息经线路 342 和 344 提供到追逆单元 191。利用被总的标识为 346 的多个多路复用器完成最小路径量度的选择。为了允许该路径量度能用 6 比特表示以便减少硬件, 一个再度量单元 348 对在加法器 350 中的最小路径量度进行再度量。按照下面的公式对该最小路径值进行再度量:

$$RV = \frac{xz^{-2}}{1 + z^{-1} + z^{-2}} \quad (2)$$

其中 x 是最小路径量度, RV 为再度量值, z^{-1} 和 z^{-2} 是被分别延迟了 1 和 2 个相加 - 比较 - 选择框 306 的工作周期的 x 。使用延迟是因为需要两个周期计算该最小路径量度。所使用的再度量功能保证再度量值 RV 将永远为正。该再度量值在总线 352 上输出。

现参考图 3, 所述维特比解码器具有一个控制框 195, 其具有几种功能。在第一工作方式中, 具有最小路径量度的路径的非法状态转移被计数作为该当前被估计收缩比率、收缩相位和载波相位是否已被正确确定的量度。根据该非法状态转移数, 对收缩比率、收缩相位和载波相位的新的组合进行挑选。如果该非法状态转移数是在一个特定容限范围内, 则开始第二工作方式, 在该方式中启动一个输出数据流, 在该数据流中寻找一个正确的同步码型。然而, 第一工作方式的最后状态被保留。因此, 如果同步没有实现, 该第一工作方式在该最后状态被恢复。通过参考图 17 可更好地对其进行理解。最初,

在步骤 354，非法状态计数器和等待计数器被复位。在判决步骤 356，立即做出检查以确定非法状态转移已被超过的允许数。

在步骤 370，接着对非法状态转移做出检测。如果非法状态转移没有发生，则控制立即转到判决步骤 360。如果一个非法状态转移发生，则在步骤 372，一个非法状态转移计数器被递增。否则控制转到步骤 358。另一非法状态转移的累积次数的检测在判决步骤 374 中执行。如果该非法状态次数仍在容限范围内，则控制转到步骤 358。否则执行步骤 336，如将在下面所描述的。

在步骤 358 使等待计数器递增。接着，在判决步骤 360，做出一个检测，根据该等待计数器的状态确定是否已对 256 个周期进行估计。如果否，控制返回到步骤 356。如果已对 256 周期做出估计，且非法状态转移保持在容限范围内，则在步骤 326 启动同步搜索。然后控制进到判决步骤 363，在该步骤对该同步单元活动进行检测。直到同步失效，控制保持在步骤 364。

终于同步失效，控制返回到在判决步骤 366 的第一工作方式。在步骤 356 的任一次执行中，如果非法状态转移的次数不在容限范围内，则控制也移到步骤 366。步骤 366 是一个判决步骤，其中对收缩相位和载波相位的所有可能的组合的殆尽做出检测。如果上述组合还没有殆尽，则载波相位在步骤 368 载波相位被改变，且控制返回到步骤 354。如果该检测在步骤 366 失效，则在判决步骤 376 做出另一个检测，以确定是否已对所有收缩比率做出估计。如果殆尽没有发生，则在步骤 378 该收缩比率和相位被改变。如果已对所有收缩比率和相位做出估计，则在步骤 380 增加容限，且控制再次返回到步骤 354。

图 17 中所示的流程图的实现在图 18 中示出。增量器 382 中的等待计数器被递增，其值被放置到总线 384 中。在组合逻辑 386 中对等待计数器进行检测。被容许非法转移的次数作为信号在总线 388 上被发出。然后在线路 394 上产生一个脉冲，在线路 396 上被输出，和经一逻辑网络 398 反馈到控制单元 400。该控制单元 400 根据参考图 17 所讨论的收缩，在总地称作 402 的线路上输出一个新的载波相位，收缩比率，收缩相位，和新的容限。当由一逻辑网络 406 进行解码将在先状态作为输入时，非法状态转移被作为信号在线路 404 上发送。在增量器 408 中计数线路 404 上的转移，新的计数值被放置在总线 410 上。

用于在数据流中寻找同步比特的第二工作方式在线路 412 上开始。该线路是一组合逻辑网络 414 的输出，该网络由几个控制信号控制，即容限检测线路 396 的状态，线路 416 上等待计数器的状态，和在线路 418 上指示的该解码器的第二工作方式的当前状态。

5 参考图 19 能够更好地理解 systolic 追逆矩阵单元 191(图 3)。追逆矩阵单元 191 被链接到路径量度产生框 189 的相加 - 比较 - 选择框 306(图 15)，其包括一系列追逆纵列 420，每个追逆纵列 420 代表在一时间点由相加 - 比较 - 选择框 306 和路径量度产生框 189 确定的所有历史上的幸存路径。每个追逆纵列 420 具有多个追逆部件 422，而每个追逆部件 422 接收追逆数据 424
10 的 m 比特。正如在这里所解释的，在本优选实施例中 $m=2$ 。根据至少一个在先追逆纵列(未示出)的内容，追逆纵列的追逆部件 424 由三个预解码选择线路 426，428 和 430 编址，因为由三个解码器 432 进行解码。每个追逆纵列 420 的输出被放置在预充电线路 434 上。

按照公知的 2 - 阶维特比解码的理论，在每个追逆纵列 420 中获得两个
15 比特以变成下一追逆纵列的两个最有效比特。在追逆中的每一级，一个 6 比特状态对 64 个位置之一编址以得到该追逆元素的内容和在该追逆中建立下一个状态。该 64 对 1 多路复用是通过对两个数据线路 434 预充电来实现。

如上所解释，选择线路 426, 428 和 430 根据在先追逆元素进行连接，线路 426 抽头对应其状态数的解码状态[1: 0]，线路 428 抽头状态[3: 2]，和
20 线路 430 抽头状态[5: 4]。在时钟 $ph1$ 436 上，两个预充电线路 434 被拉到 VDD 上。在时钟 $ph0$ 438 上，通过选择线路 426, 428 和 430 仅选择追逆部件 424 之一，和根据该追逆数据将预充电线路 434 向下拽。在锁存器 440，442 中锁存预充电线路 434 的状态，以被用作下一追逆纵列(未示出)的 2 个最有效位。已经发现，预充电线路 434 的使用大大地减小了追逆单元 191 所需的
25 面积。

当到达最后的追逆纵列时，全部被解码 I，Q 数据的两个比特在预充电线路 434 上被输出以供同步时钟 193(图 3)使用。在该追逆中的历史量(窗口的大小)和在量化的 I，Q 数据流中的级数对维特比解码器有着重要的影响。

30 维特比追逆的进一步描述

在下面的描述中，使用了 24 个追逆级实现所需的差错校正性能。然而

本发明也可使用其他具有不同级数的追逆设备。因为目前的维特比解码器被设计使用了 2 - 阶解码, 所以每级对追逆数据的 64×2 个比特起作用。现参考图 20, 其更详细地示出使用在上述设备中的追逆单元 502。其通过检索包括在寄存器 504 中的具有 6 位数据的追逆数据操作, 所述 6 位数据被存储在寄存器 506, 并表示“最可能状态”。如上所述, 输入到追逆单元 502 的“最可能状态”是这样一种状态(拥有 64 个可能性), 即其有着如由路径量度产生器 508 确定的最小路径量度。在追逆单元 502 中每级的寄存器 504 的重复对于重新定时是必要的。在追逆的后续级 510 中代表最可能状态的数据由两个选择器 512 利用在线路 514, 516 上所选追逆数据和存储在级 520 中的在先最可能状态的 4 个最高有效位 518 形成。选择器 512 由 64: 1 多路复用器实现。通过这种方法, 该电路有效地以每一时钟周期 2 阶追逆通过格形。每一追逆级的近似资源需求在表 4 中示出。

表 4

资源	计算	总计
触发器	$64 \times 2 \times 2 + 6$	262
多路复用器(64: 1)	2	2
级间连接	$64 \times 2 + 6$	134

在一种现代的综合设计中, 64: 1 多路复用器的使用遗憾地导致了更小的多路复用器的深层分级, 这些更小的多路复用器可能是太慢或尺寸太大。此外, 对追逆数据重新定时的要求导致了大量使用触发器。

现参考图 21, 图 21 示出了一个追逆单元 522, 其包括多个追逆级, 和一个特定的追逆级 524。路径量度产生器 508 的输出在 6 位总线 526 上被传递到追逆单元 522。对于当前周期具有最小路径量度的状态呈现在该 6 位总线 526 上, 其在一个 6: 64 解码器 528 中被解码。在一寄存器 530 中重新定时之后, 到达追逆级 524 的输入随后出现在一 64 位总线 532 上。该 64 位总线 532 被映射到 64 个 4: 1 多路复用器的一个组合 534 中, 根据图 2 所示的格形图, 64 个状态中的每一个对应着一个多路复用器。该格形图基本上按追逆级 524 的结构造型, 其中在组合 534 中每个多路复用器由该追逆数据设定以确定其相关状态。根据定义, 由组合 534 中每一多路复用器指示的状态表示从 4 个在前可能状态之一到达该状态的最可能阶数。来自在前级的输入追逆数据被存储在 64 位寄存器 536 中并在两个 64 位总线 538 上传递。在在前

级中 64 个状态中的每个状态的 2 比特追逆数据被映射到 64 个多路复用器的组合 534 的选择器线路 540。在当前级中被要求达到每一状态的最可能阶数在一 64 位总线 544 上被传送到后续级 542。当最小路径量度出现在 6 位总线 526 上时，其将脉动通过由追逆级 524, 542 和后续追逆级(未示出)造型的格形。这种安排的一个优点在于在所有级之间没有必要重新定时。

由于对于通过格形的每一步该“最可能状态”仅必须通过一个 4 对 1 多路复用器，所以在一个时钟周期可以执行几个追逆级，在本优选实施例中该时钟周期为 33ns。因而，仅需要在几级之后对状态和追逆数据进行重新定时，从而在设计中进一步减少了触发器的数量。在追逆单元 522 的输出端，包括总线 546 的 64 条线路被映射到 16 条线路的四个组 548，每一组随下表而定，其中所表示的状态按照图 2 所示的格形图编号。

表 5

组	代表的状态
0	0 - 15
1	16 - 31
2	32 - 47
3	48 - 63

四个组 548 被送到一个具有 4 个加法器 552 的求和单元 550 中，每个加法器对 16 个 1 位输入求和。随后该求和在 5 位总线 556 上被提供到一个 4 路比较器 554。在 2 位总线 558 上的最终输出指示到达求和单元 550 的四个输入中最小的一个(对应于最可能状态的 2 个最有效比特)，即由加法器 552 产生的 4 组 16 个状态的哪一个具有被设定为 1 的最小输出数。

例如，在每 6 个追逆级后进行重新定时时，在追逆单元 522 中每个追逆级的平均资源利用率在表 6 中给出。

表 6

资源	计算	总数
触发器	$64 \times 2 + (64 \times 2 + 64)/6$	160
多路复用器(4: 1)	64	64
互连级	$64 \times 2 + 64$	192

虽然这种设计导致了较高数量的导线连接，但这些可通过在一个超大规

模集成电路(“VLSI0”)中使用3或4层金属方便地实现。

在本发明的一个替换实施例中，加法器552的输出可以在小于隐含在本设计中的理论上的5比特带宽的总线上输出。因为在多个追逆级之后被设定为1的状态输出数应该是很小的，所以在一实际的系统中是这是可能的。

- 5 虽然参考这里所公开的结构已对本发明进行了描述，但其并不受上述细节的限制，本申请是要覆盖可能落在所附权利要求范围内的任何改进和变化。

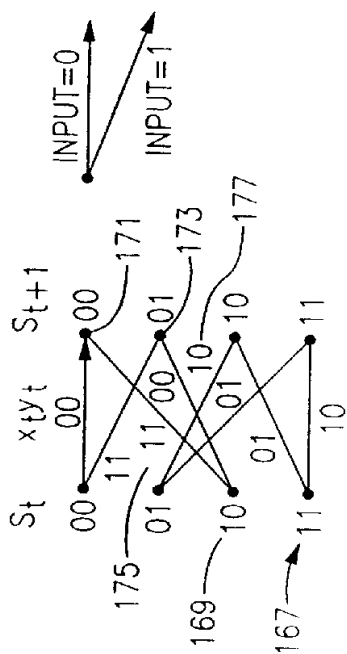


图 1

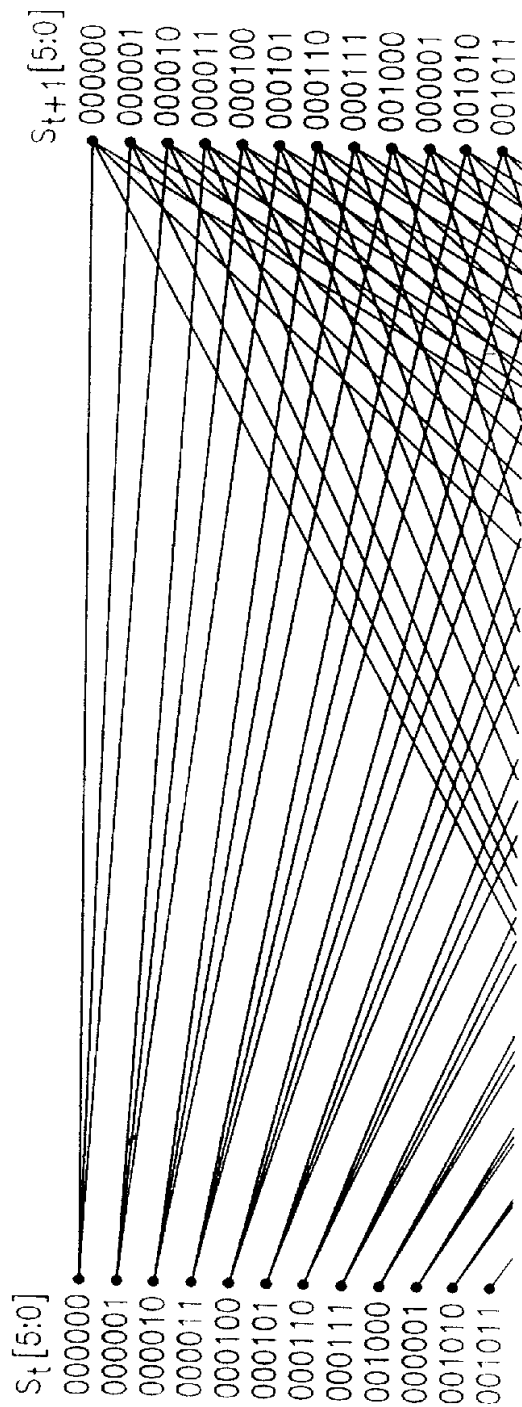


图 2

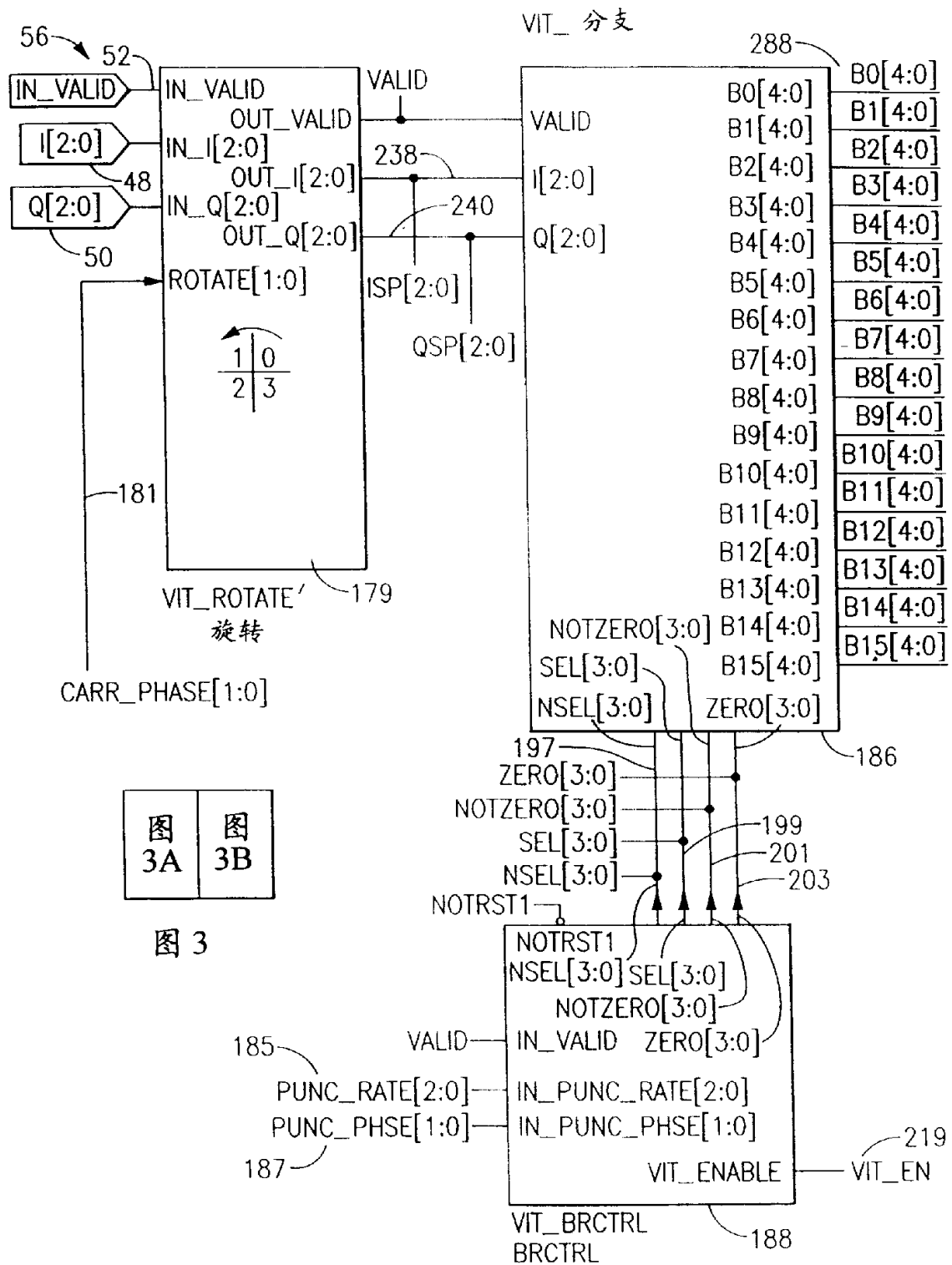


图 3A

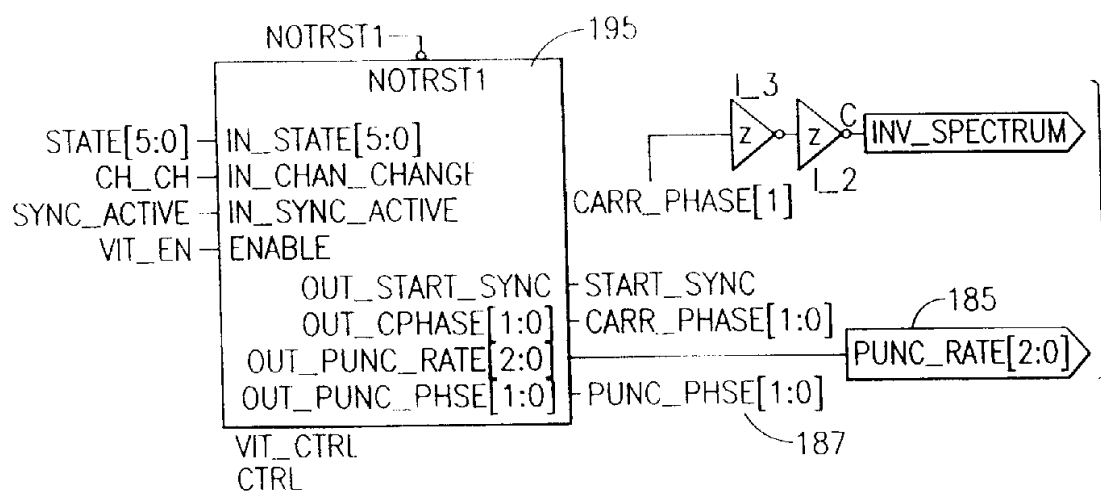
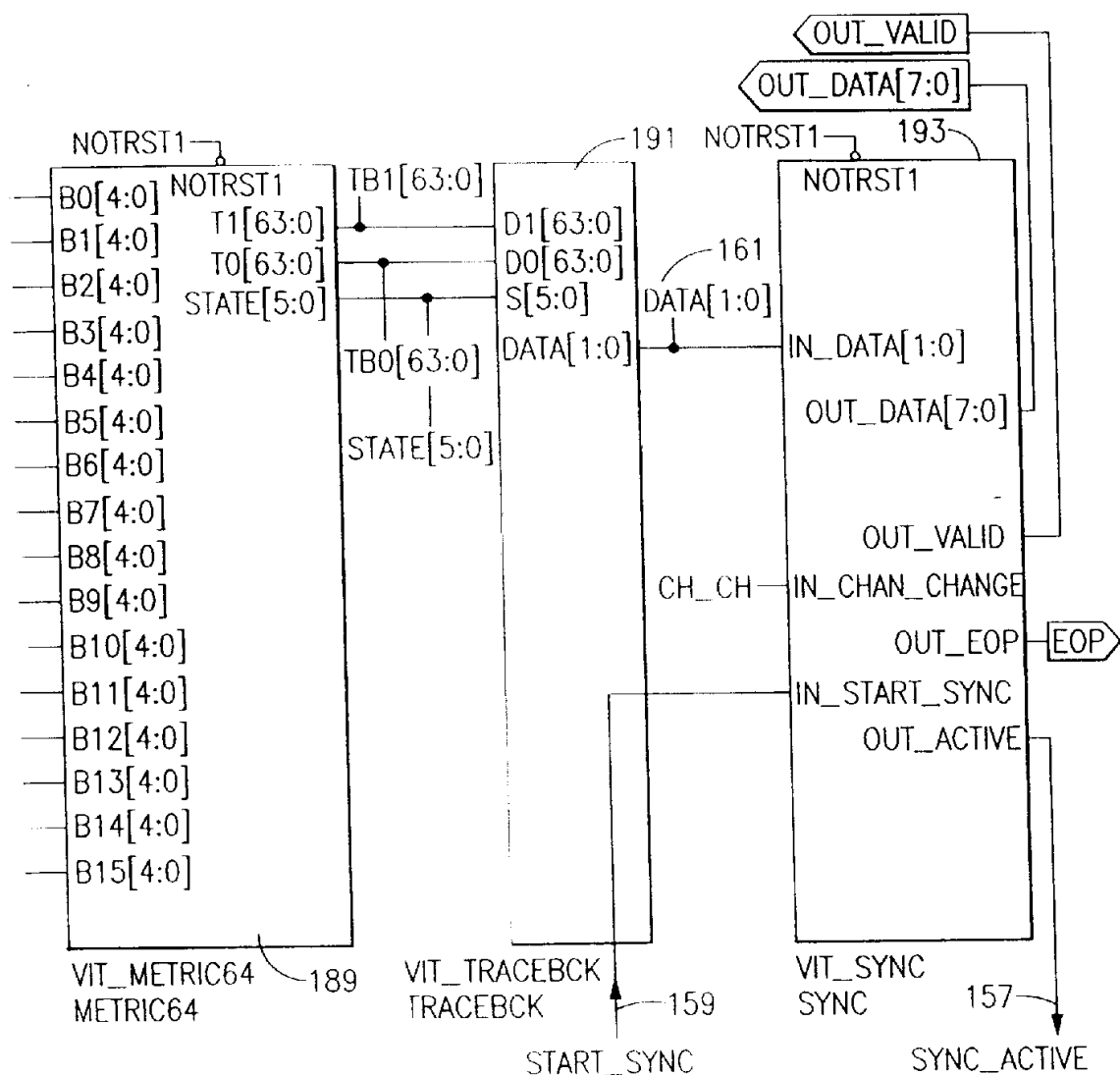


图 3B

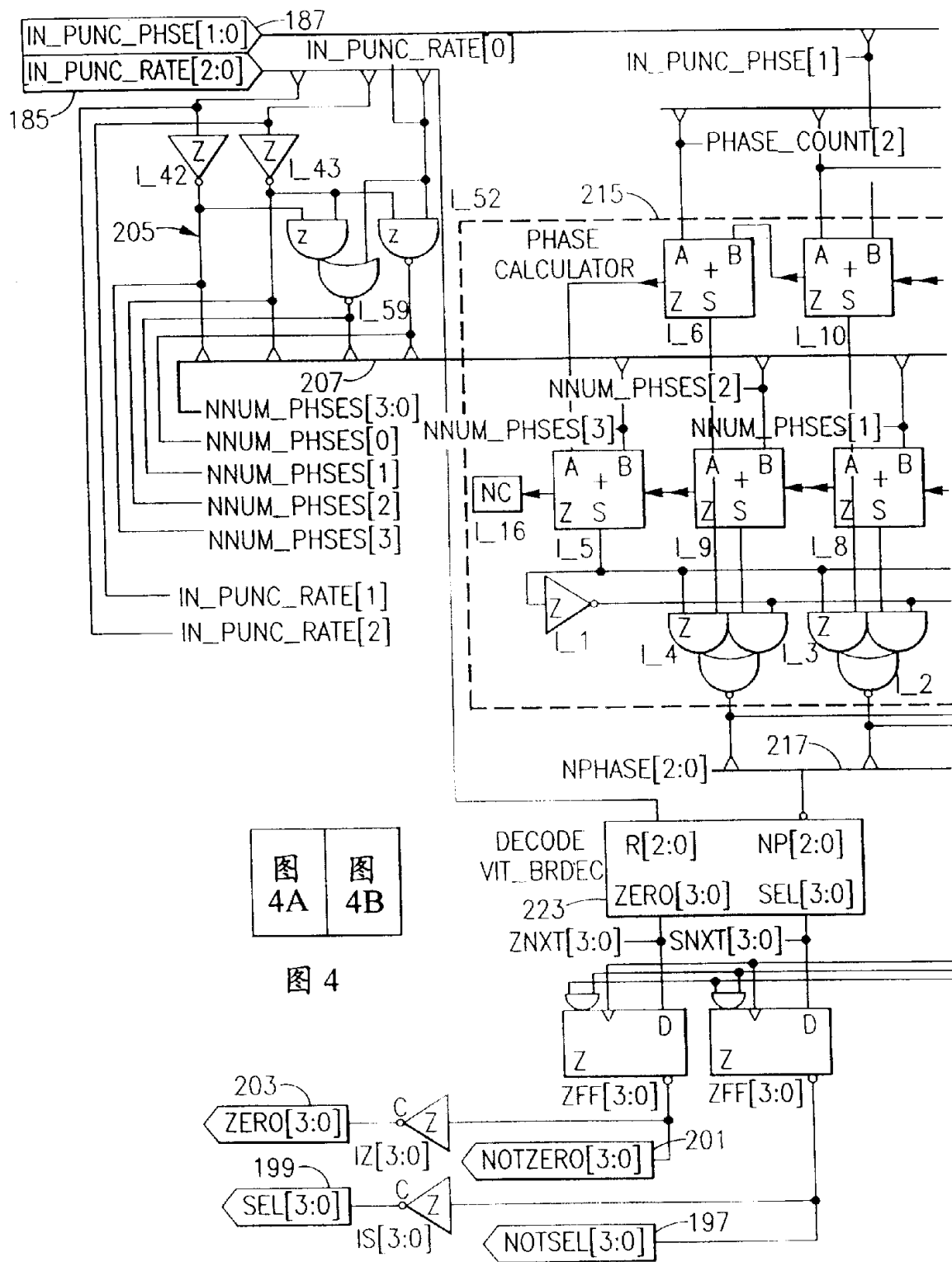


图 4A 图 4B

图 4

图 4A

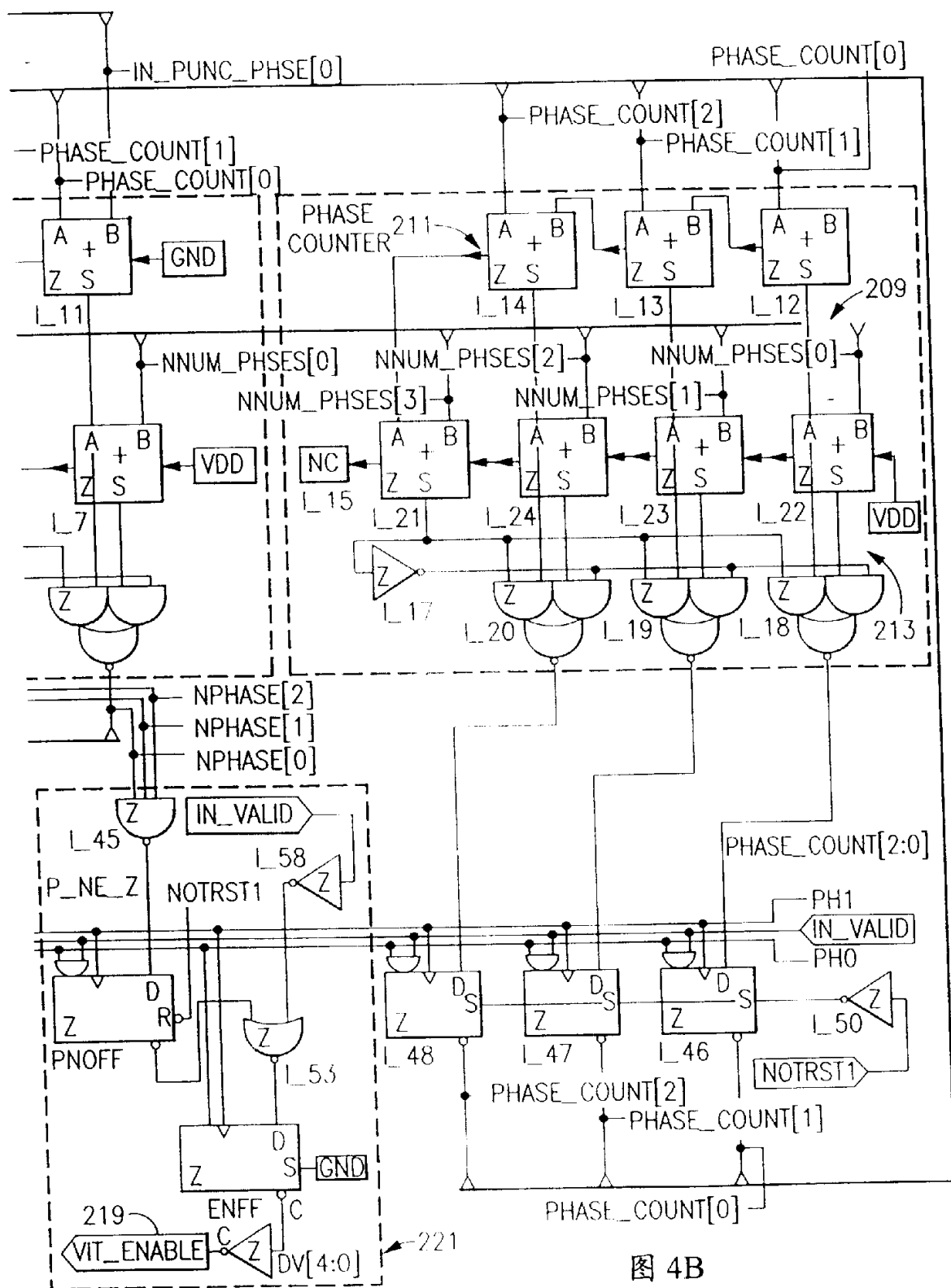
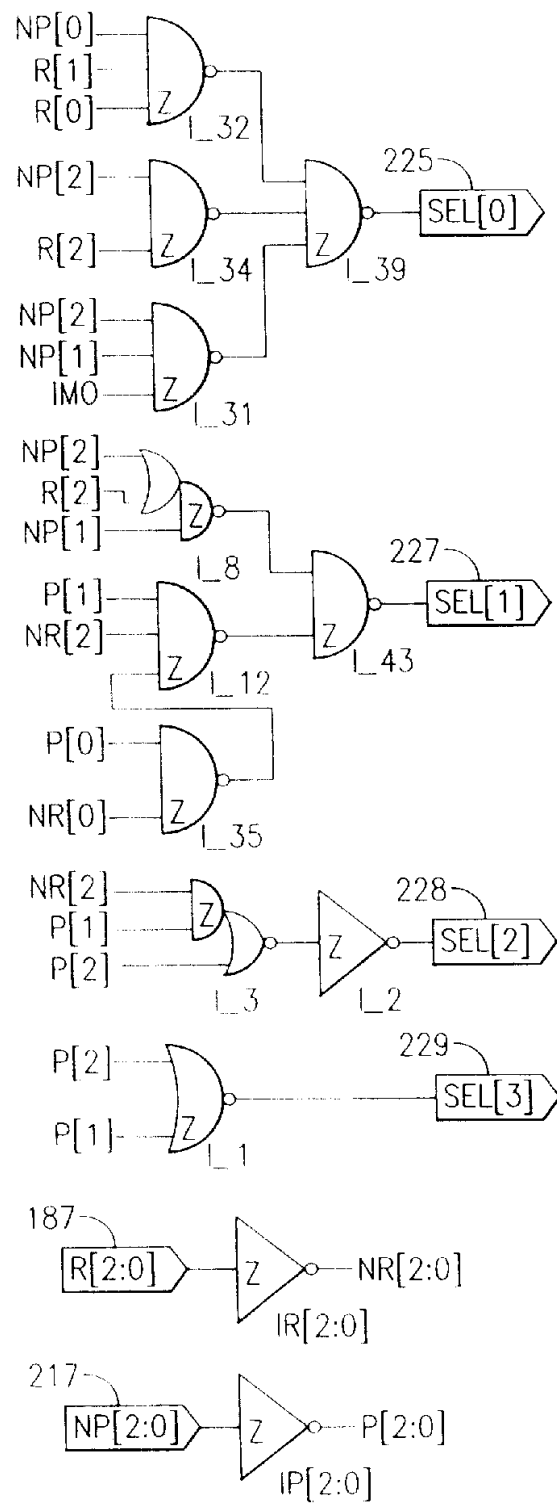


图 5



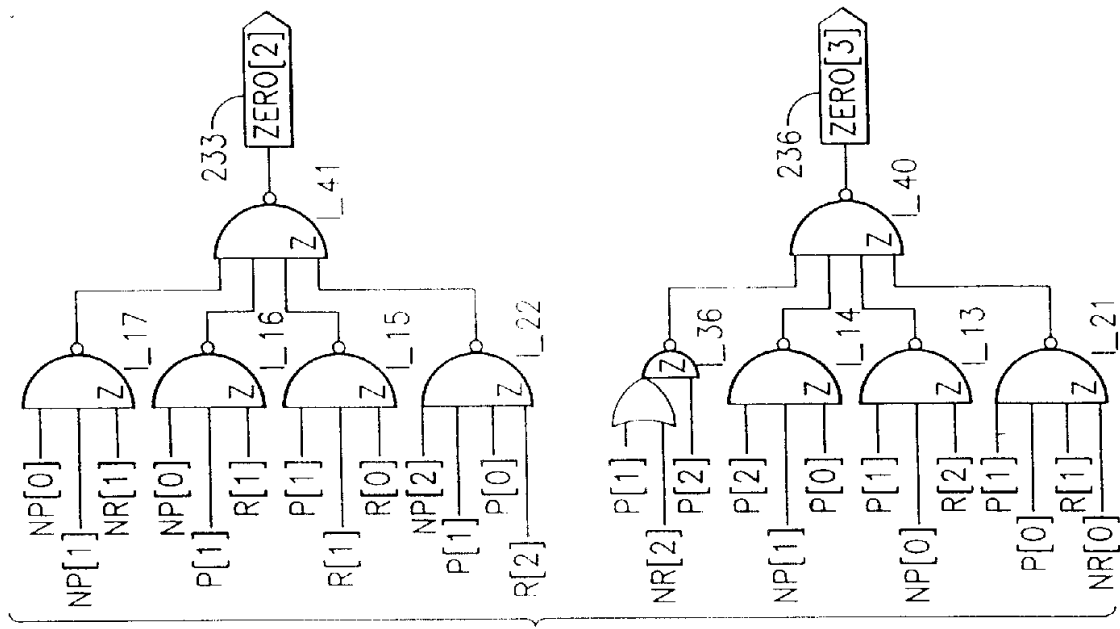


图 7

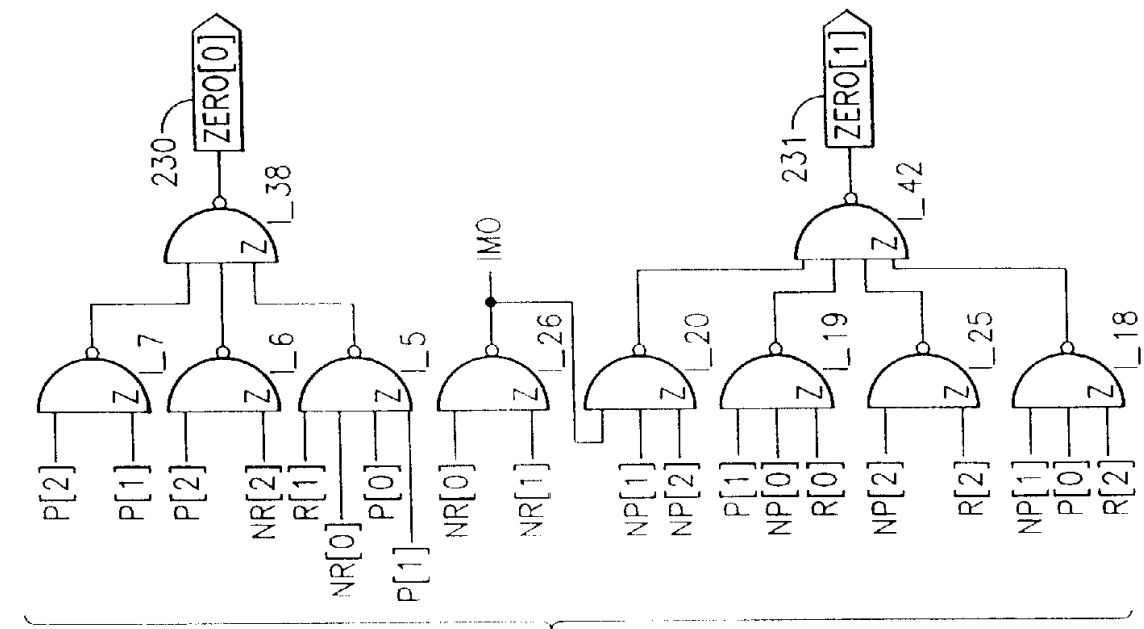
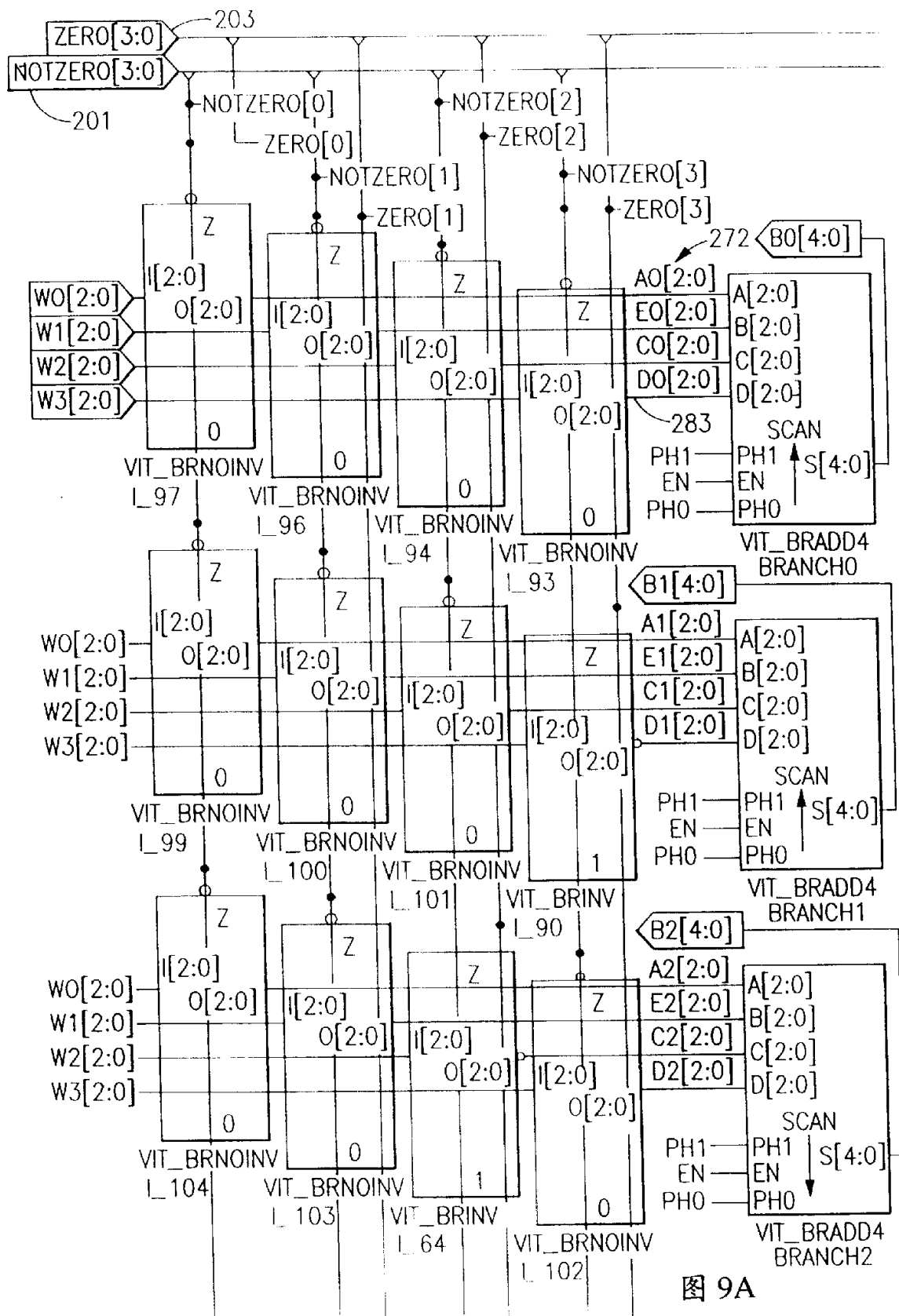


图 6



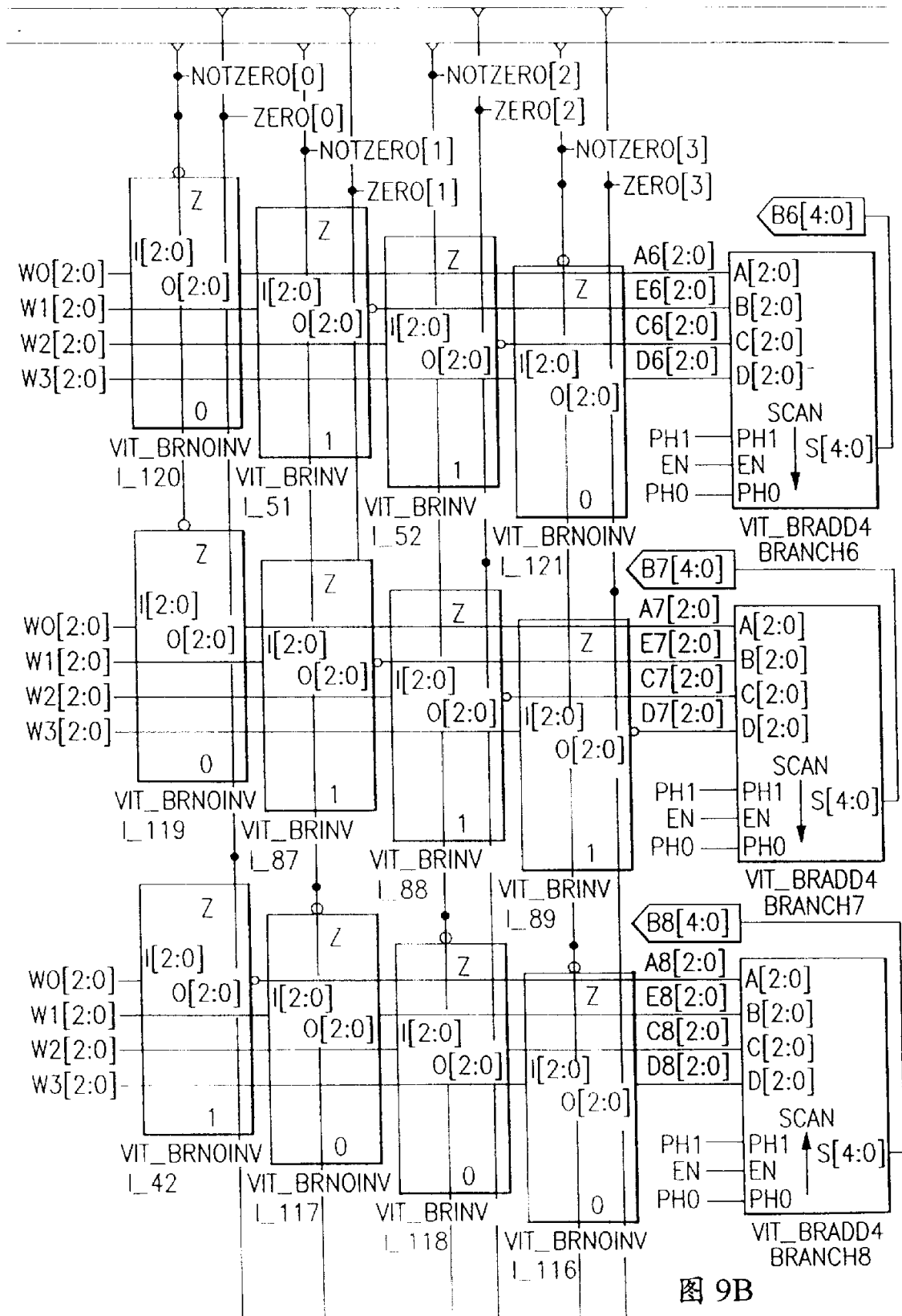


图 9B

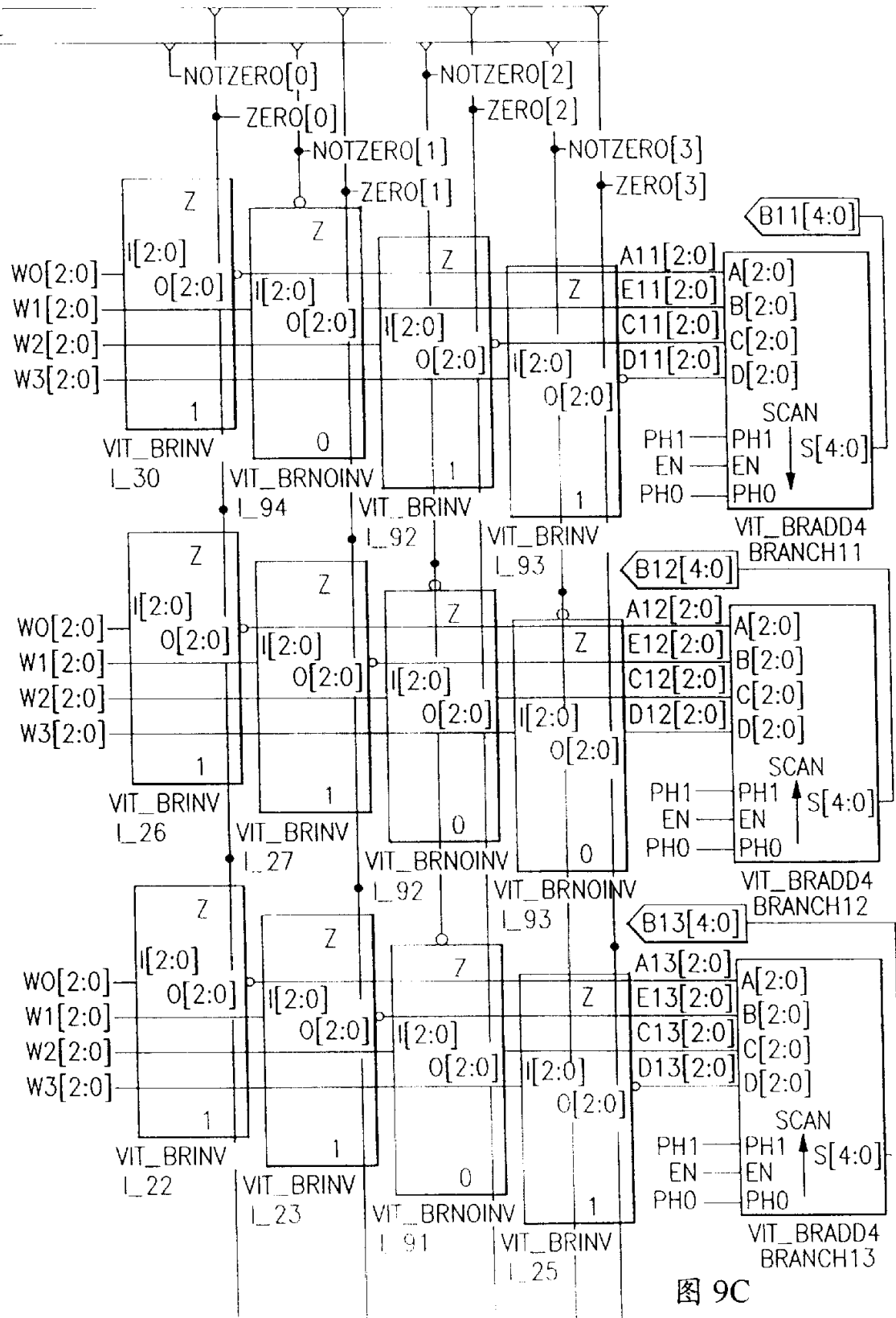
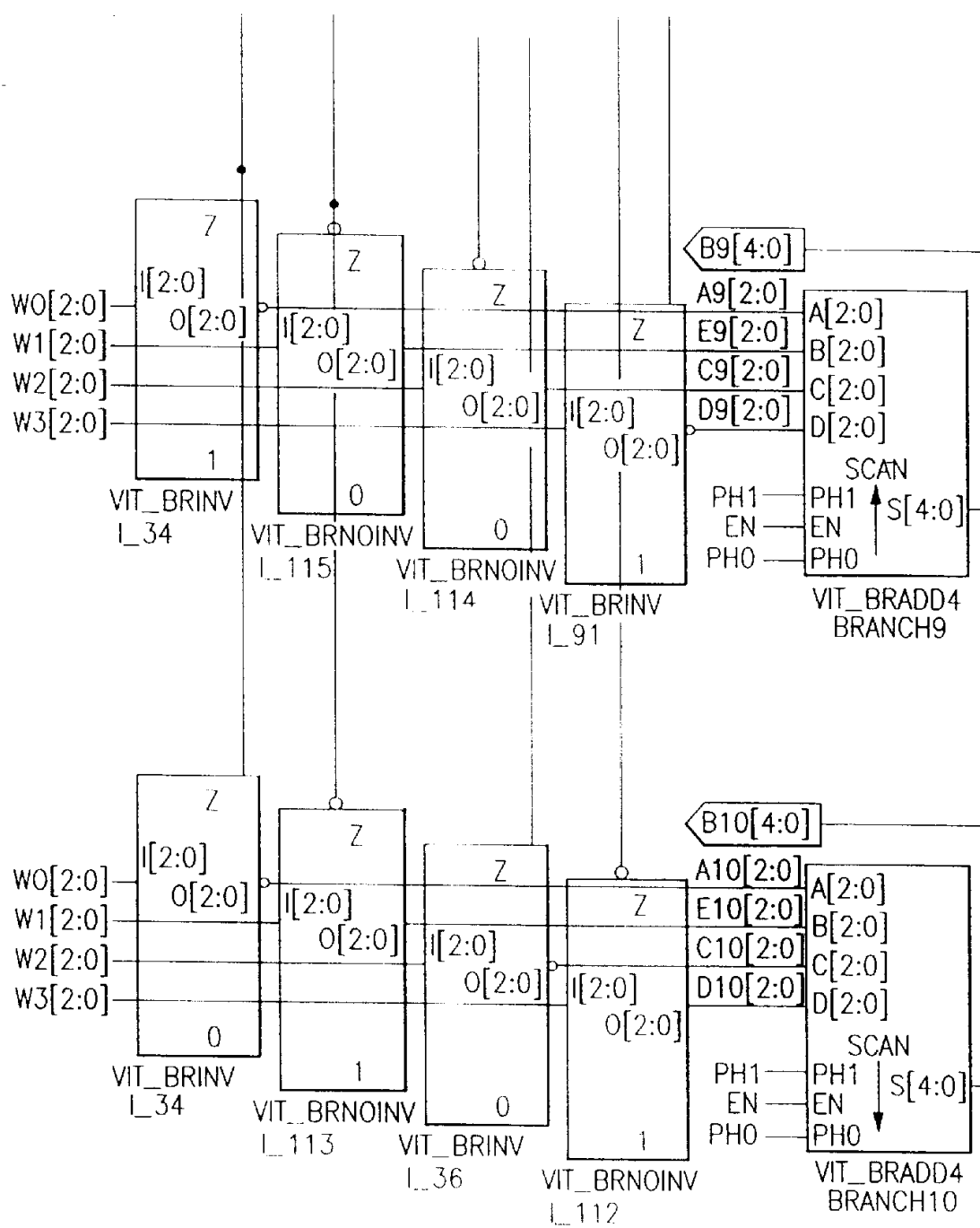


图 9C



256

图 9E

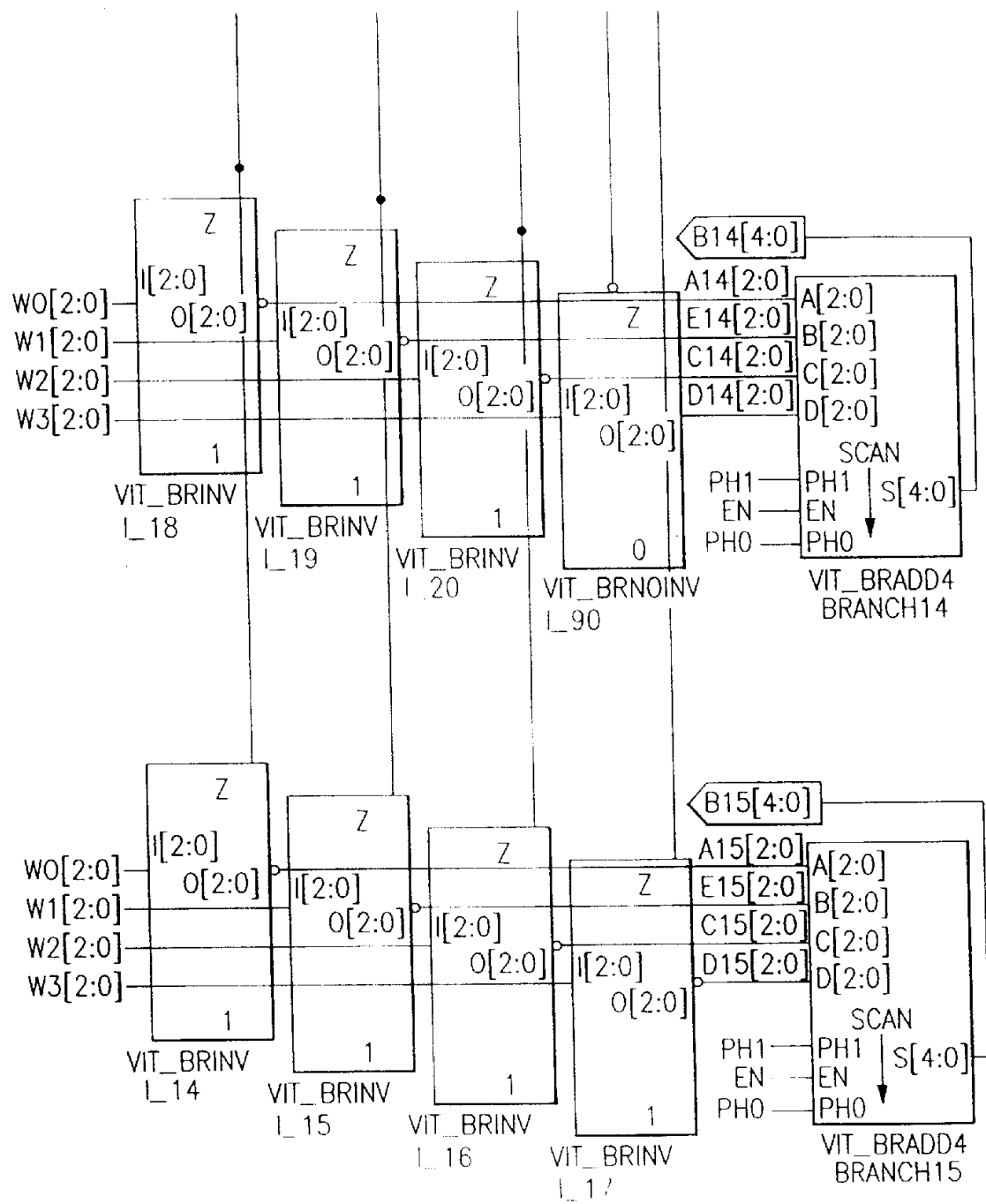


图 9F

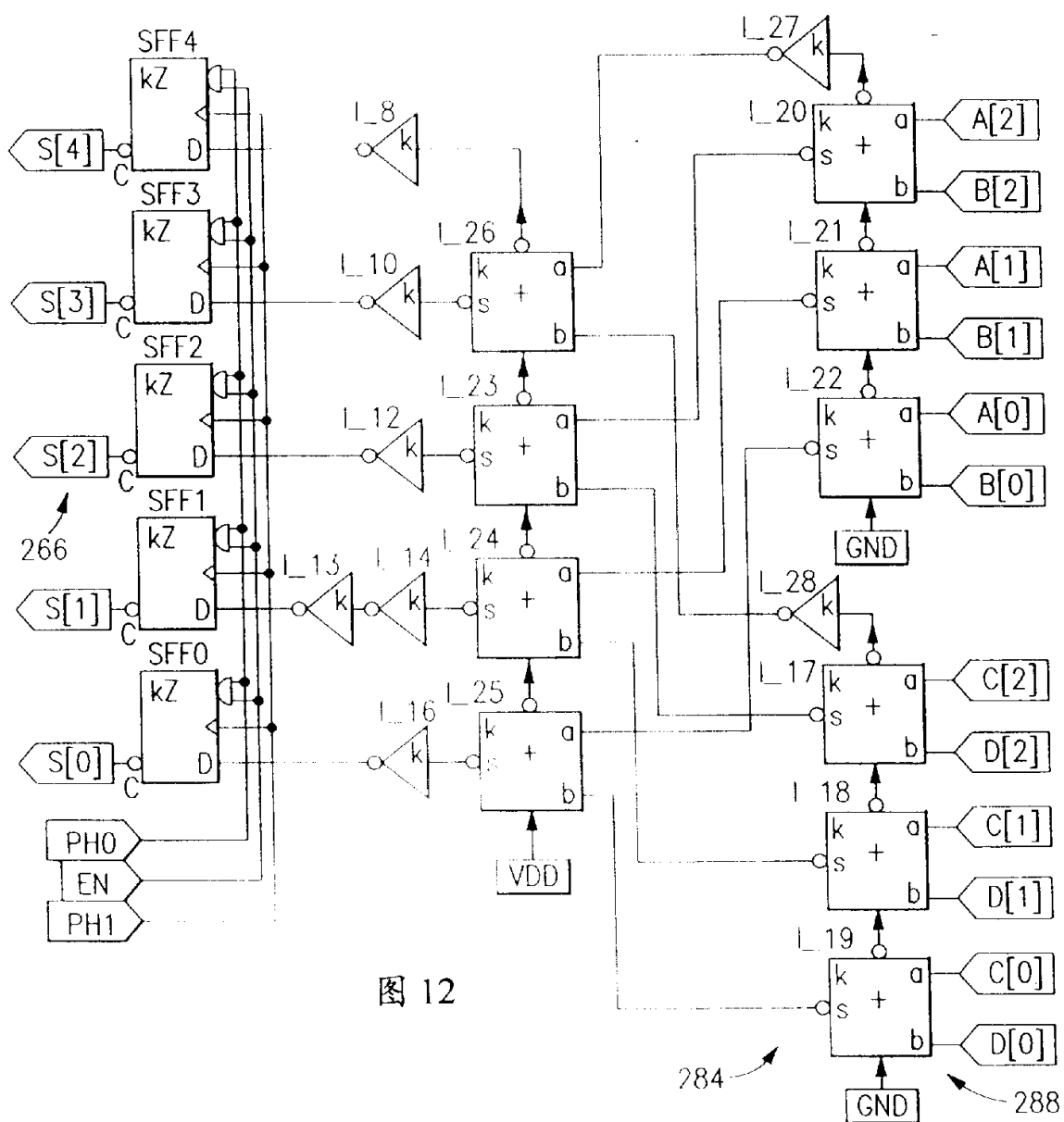
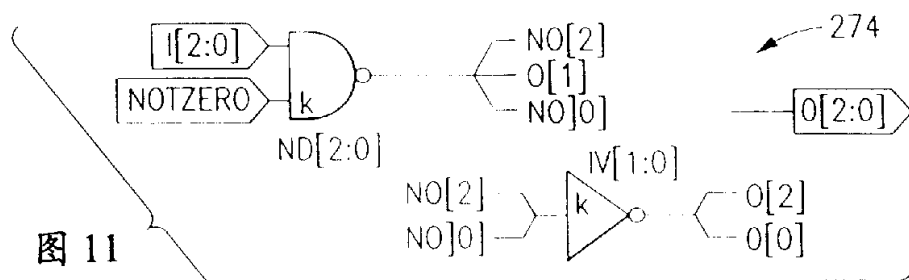
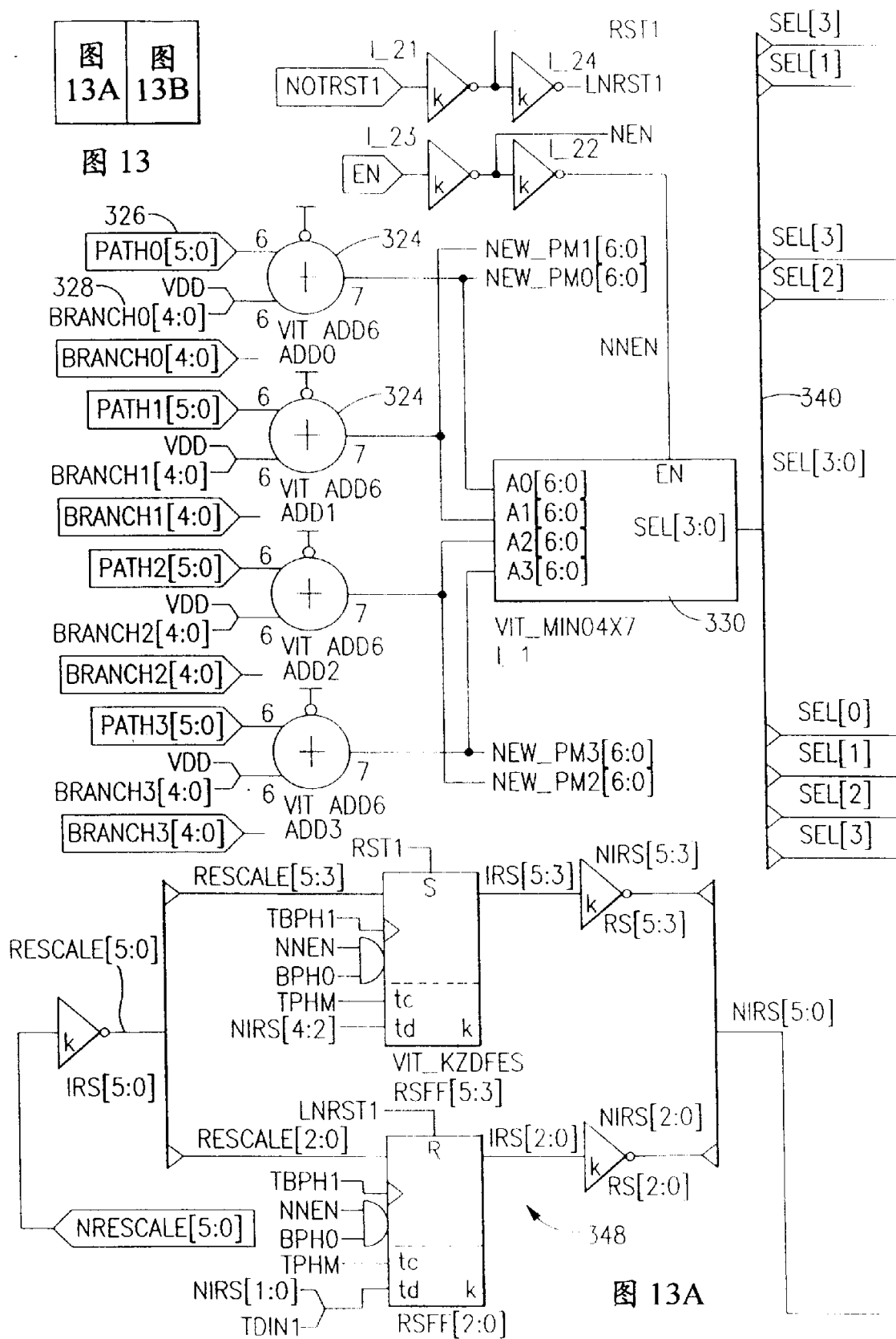


图 13A 图 13B

图 13



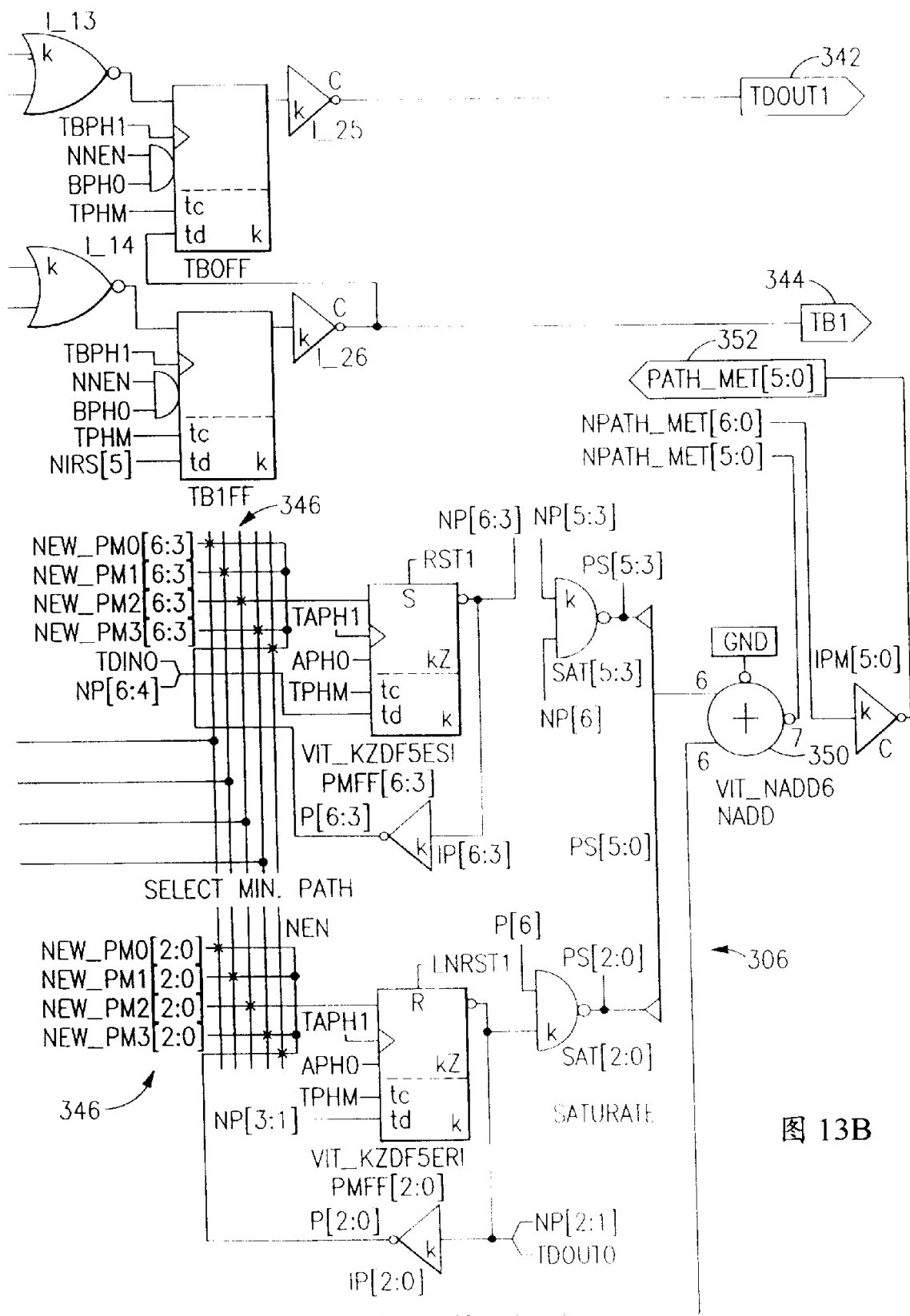


图 13B

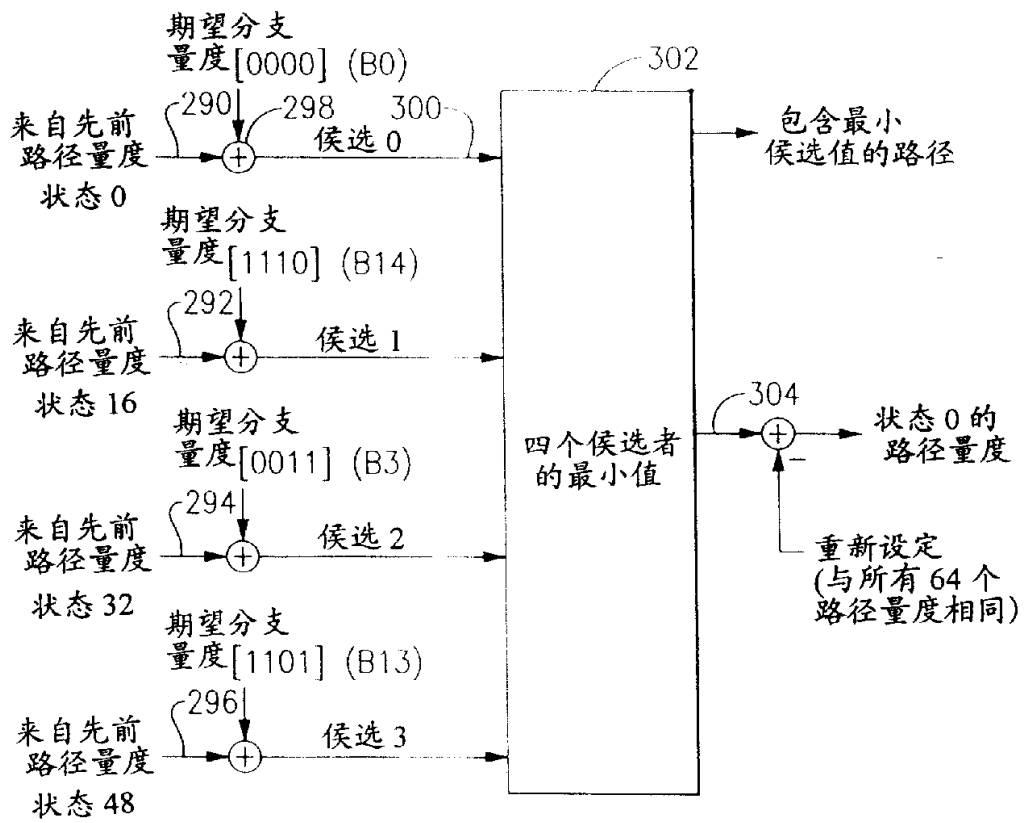
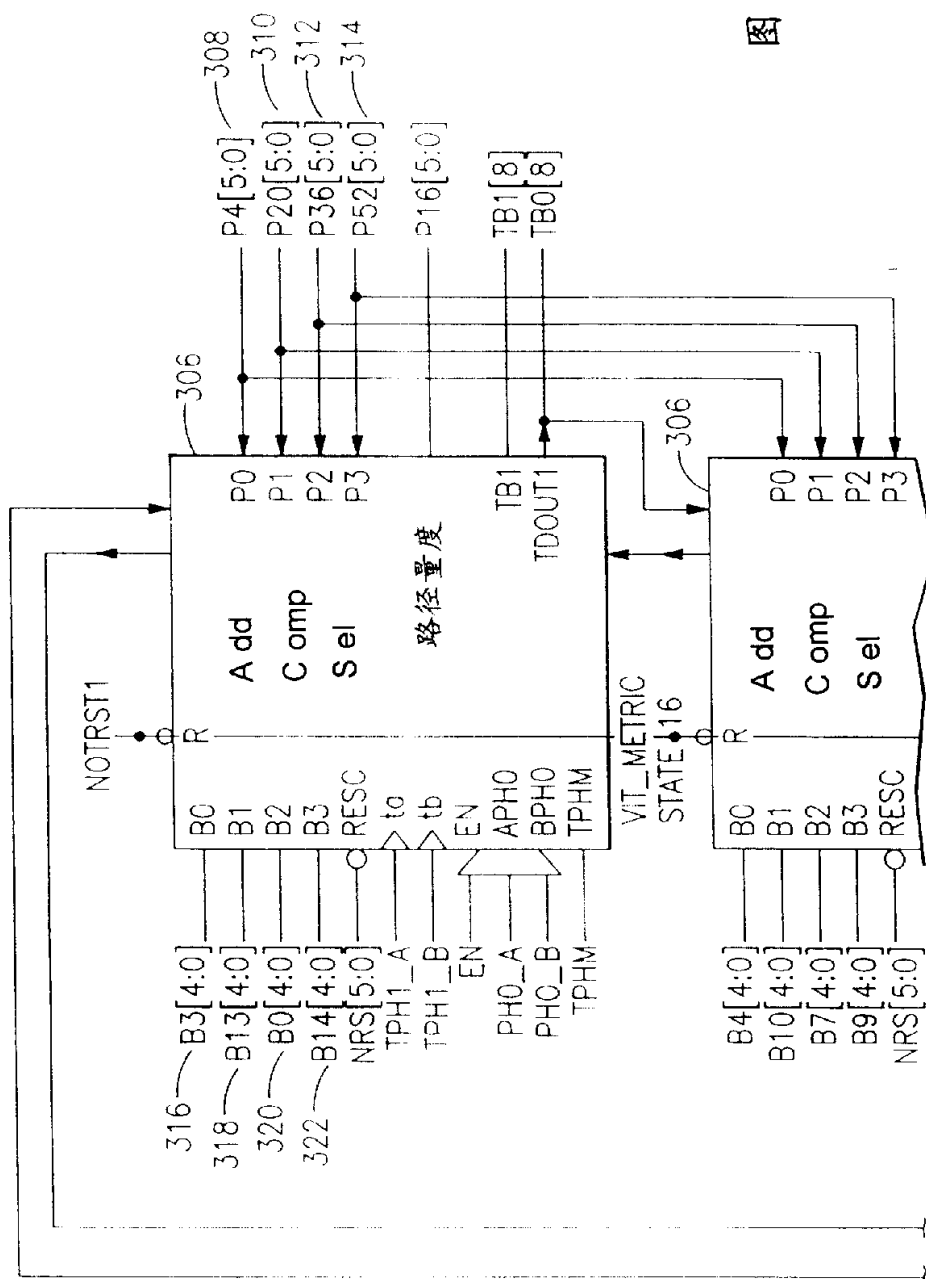
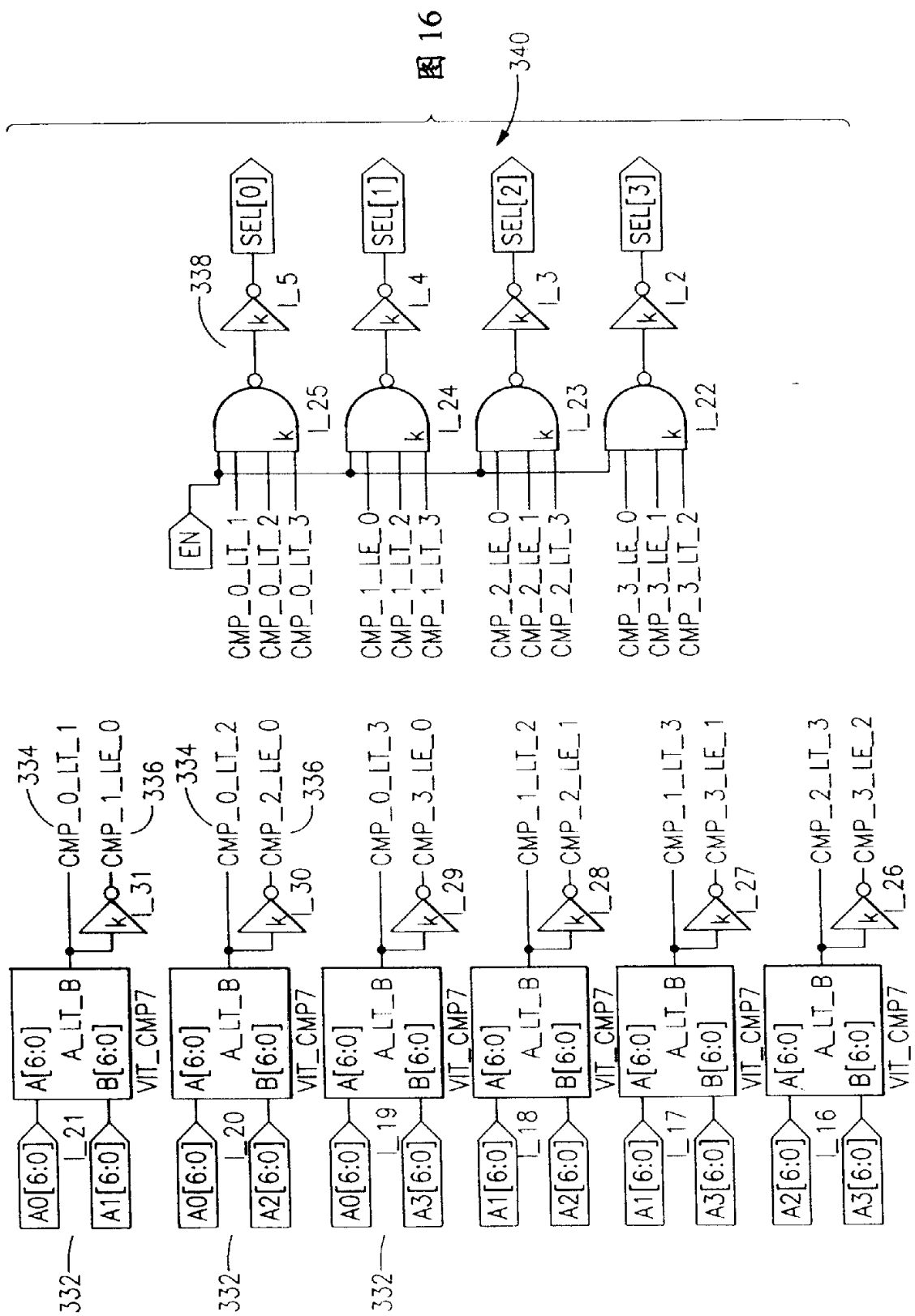


图 14





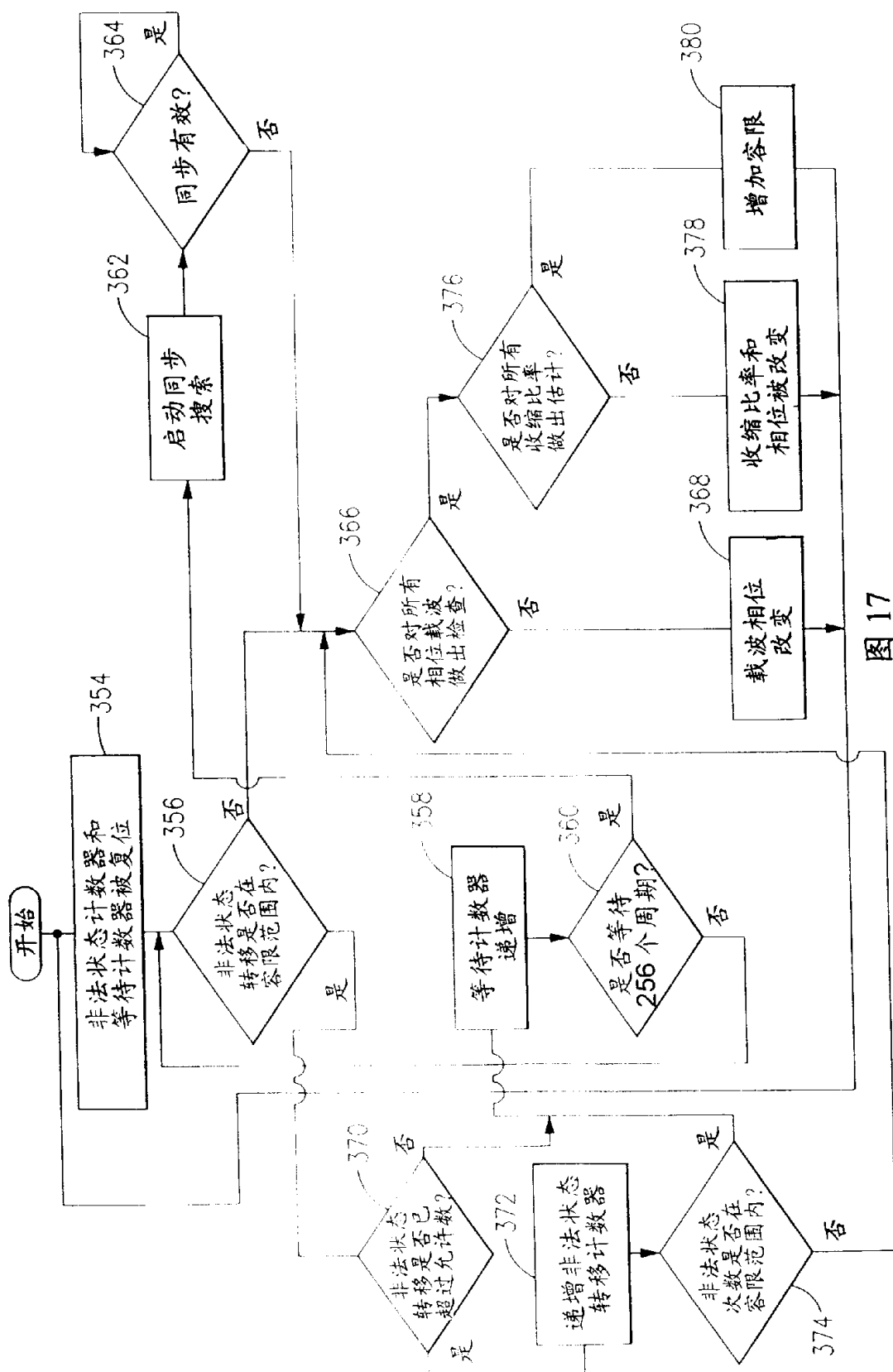
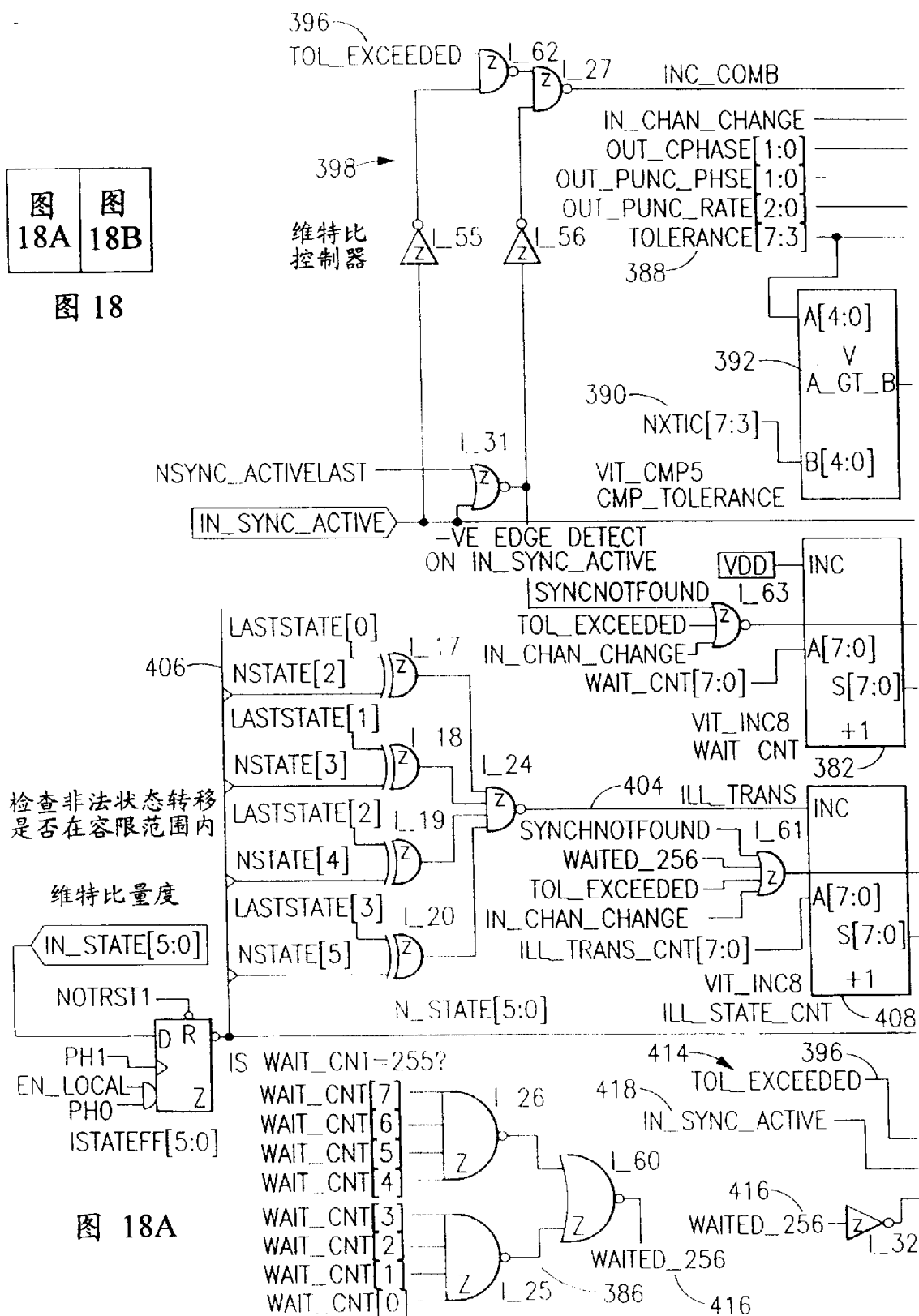


图 17

图 18A 图 18B

图 18



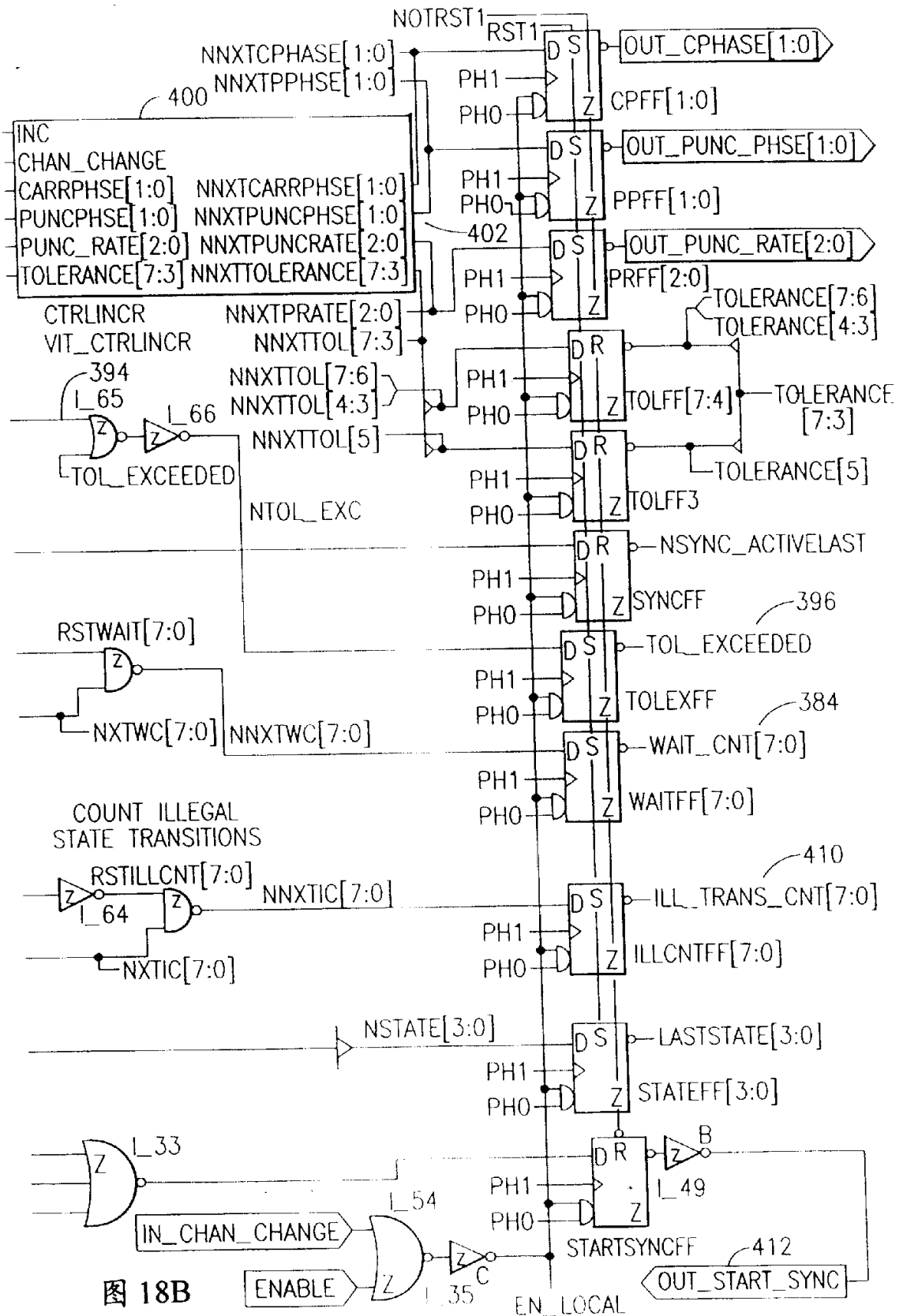


图 18B

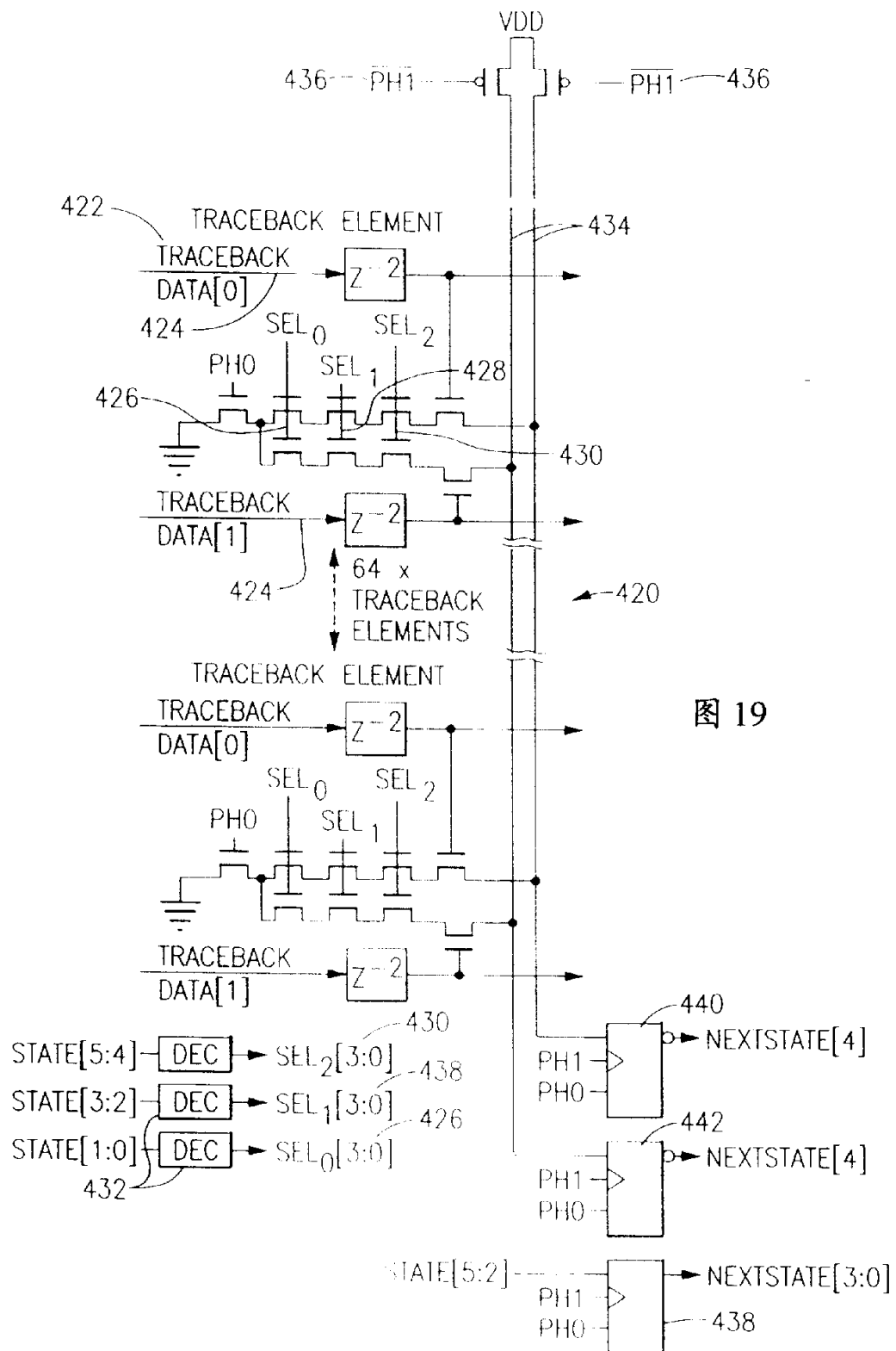


图 19

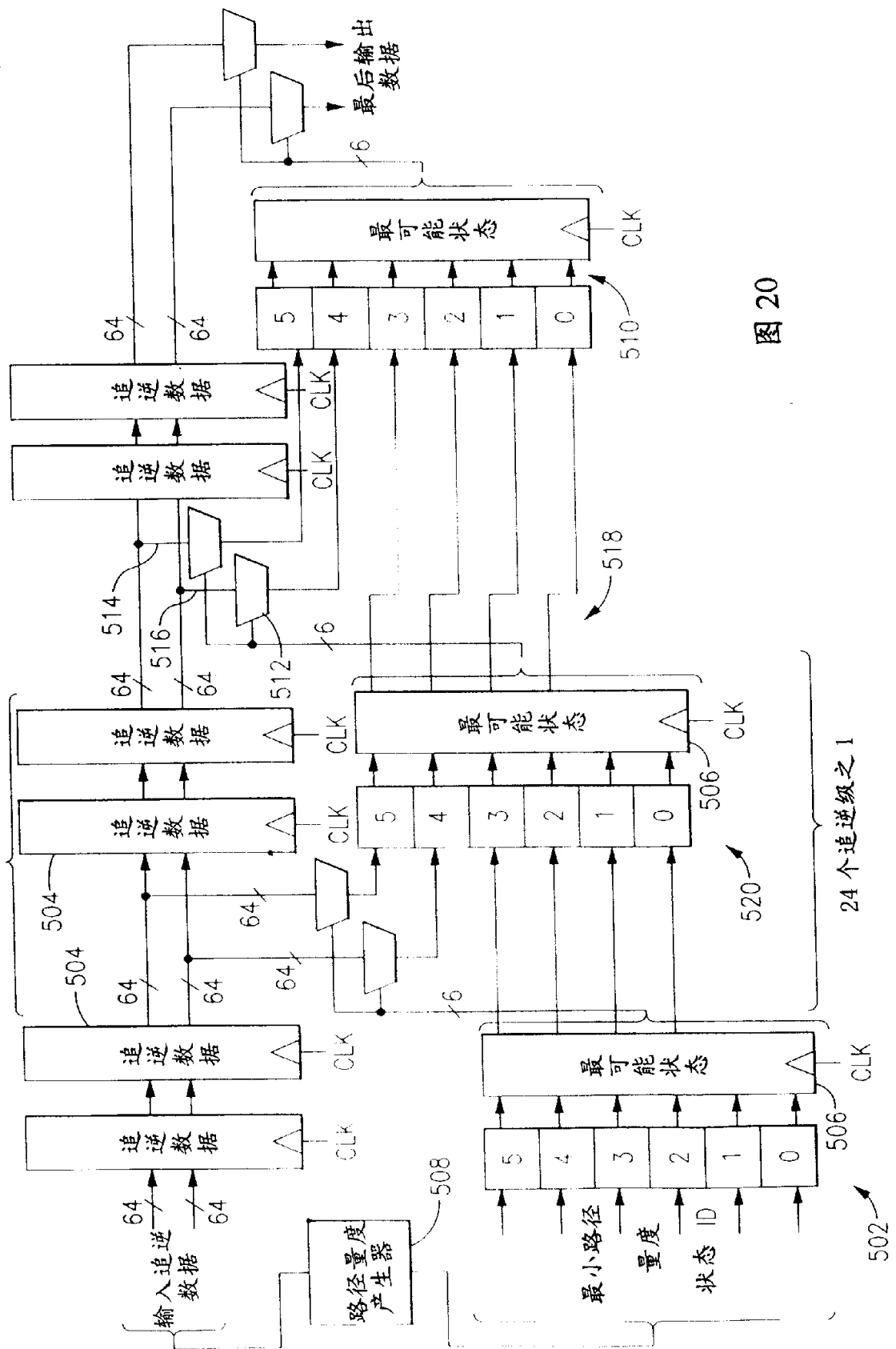


图 20

