

[12] 发明专利说明书

[21] ZL 专利号 93118166.6

[45]授权公告日 2000年2月23日

[11]授权公告号 CN 1049778C

[22]申请日 1993.9.29 [24]颁证日 1999.10.16

[21]申请号 93118166.6

[30]优先权

[32]1992.9.29 [33]JP [31]259563/1992

[32]1993.6.16 [33]JP [31]144948/1993

[32]1993.8.2 [33]JP [31]191092/1993

[73]专利权人 松下电器产业株式会社

地址 日本大阪

[72]发明人 浅野延夫 上杉充

石川利广 冈本稔

[56]参考文献

JP 平 4-352518 1992.12.7 H03M13/12

US4,979,175 1990.12.18 H03M13/12

审查员 吴兴华

[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

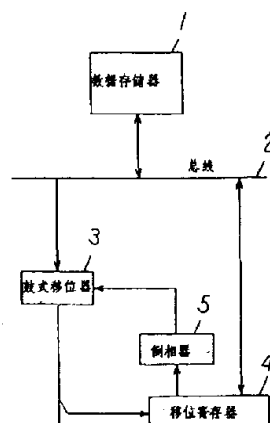
代理人 王以平

权利要求书 2 页 说明书 17 页 附图页数 10 页

[54]发明名称 用于实现误码纠错用卷积码的维特比译码的运算装置

[57]摘要

本发明公开了一种用于进行误码纠错用卷积码的维特比译码的运算装置,其包括存储器,鼓式移位器,移位寄存器及数据变换器。其中以数据变换器的输出信号数据作为鼓式移位器的移位比特数,将从所述存储器读取的数据按照由所述数据变换器的输出信号数据所指定的移位比特数,由鼓式移位器进行移位,并将鼓式移位器的输出信号数据输入到移位寄存器,从而高速地进行维特比译码。



权 利 要 求 书

1. 一种运算装置，用于误码纠错用卷积码的维特比译码，其特征在于：包括

一个存储器（1），用于存储数据；

一个鼓式移位器（3），用于将从所述存储器读出的数据移位；

一个移位寄存器（4），用于接收表示所述鼓式移位器的输出信号的数据的特定一比特的移位输入，并执行对于所述存储器的数据的读出和存储；及

一个数据变换器（5），用于将所述移位寄存器的特定寄存器输出进行变换，

其特征在于：

所述鼓式移位器（3）使用表示所述数据变换器（5）的输出信号的数据作为其一个移位比特数；并且

所述鼓式移位器（3）以由所述数据变换器（5）的输出信号数据所指定的移位比特数将从所述存储器（1）读取的数据进行移位；

所述移位寄存器（4）接收表示所述鼓式移位器（3）的输出信号的数据，从而高速度地进行维特比译码。

2. 如权利要求1所述的运算装置，其特征在于，所述数据变换器（5）包括倒相器。

3. 如权利要求1所述的运算装置，其特征在于，进一步包括：

算术逻辑运算器（8）；

多个存储装置（9，10），用于算术逻辑运算器进行运算

时使用；

第 1 寄存器 (7)，用于保持由所述鼓式移位器进行移位时的移位比特数；及

第 2 寄存器 (11, 12)，用于暂时存储所述算术逻辑运算的运算结果。

4. 如权利要求 3 所述的运算装置，其特征在于，所述数据变换器 (5) 包括倒相器。

5. 如权利要求 1 所述的运算装置，其特征在于，进一步包括：

算术逻辑运算器 (8)；

多个存储装置 (9, 10)，用于算术逻辑运算器进行运算时使用；

第 1 寄存器 (7)，用于保持由所述鼓式移位器进行移位时的移位比特数；及

第 2 寄存器 (11, 12)，用于暂时存储所述鼓式移位器的输出信号。

6. 如权利要求 5 所述的运算装置，其特征在于：所述数据变换器 (5) 包括倒相器。

说明书

用于实现误码纠错用卷积码的维特比译码的运算装置

本发明涉及数字信号处理微处理器内部的运算装置，该数字信号处理微处理器用来进行误码校正用卷积码数据的维特比译码。

近年来，数字信号处理微处理器（以下简称为 DSP）伴随移动通信领域中数字化的趋势，例如作为组装在携带式电话等机器内的微处理器受到人们注视。在移动无线回路的数据通信中，经常发生误码现象，所以，必须进行误码校正处理。对于误码校正的方法，有对卷积码使用维特比译码的方法，有时在误码校正处理中使用 DSP。

维特比译码就是利用反复进行加法运算、比较、选择这种单纯的处理和最后将数据译码的跟踪反馈操作而实现卷积码的最优译码。

在维特比译码中，每当得到与信息位 1 比特对应的编码数据（接收数据序列）时，就计算该时刻各个状态路径的汉明距（量度），求出幸存路径。

图 1 是在约束长度为 K 的卷积码编码器中，对于某一时刻的状态 S_{2n} （ n 为正整数），根据前一时刻的状态 S_n 和状态（数 1）表示状态迁移的 2 条路径延伸的情况。

[数 1]

$$S_{n+2}^{(k-1)}$$

路径量度 a 是输入于状态 S_{2n} 的路径输出代码与接收数据序列的汉明距（支路量度）以及到前一时刻的状态 S_n 为止的幸存路径的分支路量度总和，即路径量之和。同样，路径量度 b 是输入状态 S_{2n} 的路径输出符号与接收数据序列的汉明距（支路量度）以及到前一时刻的状态（数 1）为止的幸存路径的支路量度总和即路径量度之和。

比较输入状态 S_{2n} 的路径量度 a 和 b ，选择小的一方作为幸存路径。这样，在各个时刻，对 2^{k-1} 个状态进行为了求路径量度的加法运算、路径量度的比较和路径的选择等各种处理。另外，在路径的选择中，必须将选择哪一方的滞后作为路径选择信号

$PS[S_i], (i=0 \sim (2^{k-1} - 1))$ 预先保留。如果选择的路径的前一状态的下标小于另一方时，令 $PS[S_i] = 0$ ，否则，就令 $PS[S_i] = 1$ 。最后利用跟踪反馈进行译码时，根据该路径选择信号，在追溯幸存路径的同时，进行译码。

图 2 是对某一时刻的状态 S_{2n} (n 是正整数) 以路径选择信号 $PS[S_{2n}]$ 为基码向前一时刻的状态 S_n 或状态（数 1）追溯路径的情况。

通常，状态 S_i 和路径选择信号 $PS[S_i]$ 前一时刻的状态用 [数 2] 表示。

[数 2]

$$S \cdot i/2 + PS[S_i] \cdot 2^{(k-1)}$$

这时，在末位使用终端的卷积码时，可以以路径选择信号作为译码数据。

在图 3 中，1 是数据存储器，用来存储路径量度、路径选择信号和译码数据等。3 是鼓式移位器，使从数据存储器 1 中读取的数据移位。6 是第 1 总线，与数据存储器 1 连接，进行数据的供给及运算结果的存储等。7 是第 1 寄存器，用来保持鼓式移位器 3 进行移位时的移位比特数。8 是算术逻辑运算电路（以下简称 ALU），进行算术逻辑运算。9 是第 1 锁存器，用来暂时存储 ALU8 左侧输入的值。10 是第 2 锁存器，用来暂时存储 ALU8 右侧输入的值。11 和 12 分别是第 2 寄存器，由多个构成，用来暂时存储运算结果。13 是第 2 总线，从寄存器 11 或寄存器 12 进行数据供给。用 2 的补码体系表示鼓式移位器 3 的移位比特数，正数时为右移位，负数时为左移位。

对于上述结构的运算装置，对在以末位终止的卷积编码的编码数据进行维特比译码时跟踪反馈操作的动作进行说明。假定条件是卷积码的约束长度为 K ，进行了编码的信息位数为 n ，数据存储器 1、第 1 总线 6、第 2 总线 13、第 1 锁存器 9、第 2 锁存器 10、ALU8、第 2 寄存器 11 和 12 的位宽分别为 2^{k-1} 比特。另外，设 t 时刻的路径选择信号 $PS_t[S_i](t=0 \sim ((n+1)+(k-1), i=0 \sim (2^{k-1}-1))$ 如路径量度（数 3）那样，填在 1 个字内，以 $PM[t](t=0 \sim (n-1)+(k-1))$ 的形式存储在数据存储器中。

[数 3]

$$PM[t] = (PS_t[S_2^{(k-1)}-1], PS_t[S_2^{(k-1)}-2]$$

$$\dots, PS_t[S_1], PS_t[S_0]$$

经过译码的数据 $Y[i](i=0 \sim (n-1))$ 以 1 位 1 字的形式存储到数据存储器中。

下面，分阶段说明跟踪反馈的动作。

(1) 将固定值[0]存储到第2锁存器10内。ALU8直接将第2锁存器10的值存储到第2寄存器11中。

[从0状态开始]

对后面的阶段(2)~(10)，在 $((n-1)+(k-1))-(k-1)$ 内，对*i*反复*n*次进行减法计数。

(2) 通过第2总线13将第2寄存器11的值存储到第1锁存器9内。利用ALU8求出第1锁存器9的值的2的补码，并存储到第2寄存器12中。

(3) 通过第1总线6将第2寄存器12的值存储到第1寄存器7中。

[成为用于选择如下路径选择信号的移位比特数]

(4) 从数据存储器1中读出路径量度 $PM[i]$ ，利用鼓式移位器3移位第1寄存器7指示的移位比特数，然后存储到第2锁存器10内。ALU8直接将第2锁存器10的值存储到第2寄存器12中。

[将选择的路径选择信号寄存在最低位比特(LSB)]

(5) 通过第2总线13将第2寄存器12的值存储到第1锁存器9内，将固定值[1]存储到第2锁存器10内。在ALU8中，求第1锁存器9和第2锁存器10的逻辑积，并存储到第2寄存器12内。

[只抽出LSB]

(6) 将第2寄存器12的值作为译码数据 $Y[i-(k-1)]$ 存储到数据存储器中。

[LSB成为译码数据]

(7) 将固定值 $[k]$ 存储到第1寄存器7内。

(8) 通过第2总线13将第2寄存器12的值存储到第1锁存器9内，利用鼓式移位器3把第2寄存器12的值按第1寄存器7指定的移位比特数移位，并通过第1总线6将其输出，存储到第2锁存器10内。在ALU8中，求第1锁存器9和第2锁存器10的逻辑和，并存储到第2寄存器12内。

(9) 将固定值 $[-1]$ 存储到第1寄存器7内。

(10) 用鼓式移位器3把第2寄存器12的值按第1寄存器7指定的移位比特数移位，再将其输出存储到第2锁存器10内。

ALU8直接将第2锁存器10的值存储到第2寄存器12内。

[在步骤(6)~(10)计算前一刻的状态]

这样，在上述以往的运算装置中，通过将鼓式移位器3和ALU8相结合进行运算，可以将 n 比特信息位的维特比译码中的跟踪反馈操作进行到 $(9n+1)$ 步。

另外，运算位宽小于状态数 2^{k-1} 时，或不是以末位的比特终止而连续使用卷积码时，同样也可以进行跟踪反馈操作。

另外，US4,979,175公开了一种ACS运算，通过该运算处理，从其先前旧路径度量和分支度量可以得到一个新路径度量。其中该ACS运算是用于实现维特比译码的处理之一，在该公开技术中，路径度量被定义为状态度量。在进行ACS运算时，需要用于存储旧路径度量的存储空间及用于存储计算出的新路径度量的存储空间。在该公开文件中，提供了一公共存储空间来存储新路径度量与旧路径度量。

但是，在上述以往的运算装置中，为了求前一刻的状态所

需要的运算步骤多，另外，由于以1比特1字节的形式存储译码数据，所以，要求数据存储器的容量大。

本发明就是为解决上述问题提出的，其目的旨在提供一种优异的运算装置，可以用很少的运算步骤和很少的数据存储器进行维特比译码中的跟踪反馈操作。

为了达到上述目的，本发明设有存储器、鼓式移位器、移位寄存器和数据变换器，存储器用来存储数据；鼓式移位器将从上述存储器读出的上述数据进行移位；移位寄存器也具有将上述鼓式移位器的输出信号即数据的特定的1位作为移位输入后，并向上述存储器的数据负载及上述存储器进行数据存储的功能；数据变换器用来使上述移位寄存器的特定寄存器的输出变换。以上述数据变换器的输出信号即数据作为上述鼓式移位器的移位比特数，将从上述存储器读出的上述数据按照由上述数据变换器的输出即数据所表示的移位比特数由鼓式移位器进行移位，同时，将上述鼓式移位器的输出信号即数据输入上述移位寄存器。

因此，按照本发明，通过利用鼓式移位器将从数据存储器读出的路径量度进行移位，只将所选择的路径选择信号输入移位寄存器，便可求出前一刻的状态和决定下一个鼓式移位器的移位比特数，进而可以将存储在移位寄存器中的路径选择信号作为译码数据，所以，利用很少的运算步骤和很少的数据存储器就可以进行维特比译码中的跟踪反馈操作。

图1是表示维特比译码中卷积编码器的状态转移路径的状态转移图（结构线图）。

图2是表示追溯维特比译码中跟踪反馈时的路径的状态迁移

图 (结构线图)。

图 3 是表示先有的维特比译码用运算装置的结构框图。

图 4 是本发明的一个实施例中维特比译码用运算装置的结构框图。

图 5 是其它实施例中维特比译码用运算装置的结构框图。

图 6 也是其它实施例中维特比译码用运算装置的结构框图。

图 7 是本发明的一个实施例中运算装置的结构框图。

图 8 是表示数据存储器区域分配例子的模式图，用于运算装置的动作说明。

图 9 是表示所有的状态从前一时刻的状态向现时刻的状态转移的情况的状态转移图，用于运算装置的动作说明。

图 10 是表示本发明的运算装置实施例的结构框图。

图 4 是本发明的一个实施例中维特比译码用运算装置的结构框图。

在图 4 中，1 是数据存储器，用来存储路径量度、路径选择信号和译码数据等。2 是总线，和数据存储器 1 相连接，进行数据的供给和运算结果的存储等。3 是鼓式移位器，用来移位从数据存储器 1 读取的数据。4 是移位寄存器，可以移位输入鼓式移位器 3 所输出的特定 1 比特，并且通过总线 2 向数据存储器 1 的数据负载或数据存储器 1 进行数据存储。5 是倒相器，用来使移位寄存器 4 的特定寄存器的值倒相后将移位比特数供给鼓式移位器 3。需要指出的是，倒相器 5 亦可以是其他的含有倒相器功能的数据变换器，在本发明中为了便于说明，特以倒相器为例。鼓式移位器 3 的移位比特数用 2 的补码体系表示，正数时为右移，

负数时为左移。移位寄存器 4 以移位输入端为最高位 (MSB)。

对于上述结构的运算装置,对以末位终止的卷积编码的编码数据进行维特比译码时跟踪反馈操作的动作进行说明。假定条件是卷积码的约束长度为 K ,进行过编码的信息位数为 n ,数据存储器 1、总线 2 和移位寄存器 4 的位宽为 2^{k-1} 比特。对于移位寄存器 4 的移位输入,输入鼓式移位器 3 所输出的 MSB,倒相器 5 使移位寄存器 4 的上位 ($k-1$) 比特倒相,使在该输出 ($k-1$) 比特 MSB 附加了 0 的 k 位成为鼓式移位器 3 的移位比特数。此外, t 时刻的路径选择信号 $PS_t[S_i]$ ($t = 0 \sim ((n-1)+(k-1)), i=0 \sim (2^{k-1}-1)$) 像路径量度 (数 3) 那样,填在 1 个字内,以 $PM[t]$ ($t = 0 \sim ((n-1)+(k-1))$), 的形式存储到数据存储器 1 内。

经过译码的数据 $Y[i]$ ($i=0 \sim (n-1)$) 将 2^{K-1} 比特填在 1 个字内,存储到数据存储器 1 内。下面,分步骤说明跟踪反馈的动作。

(1) 将固定值 [1] 存储到移位寄存器 4 内。

[从 0 状态开始]

对于后面的步骤 (2) ~ (3), 按照 $((n-1)+(k-1) - (k-1))$ 减法计数 i , 反复 n 次。

(2) 从数据存储器 1 读出路径量度 $PM[i]$, 由鼓式移位器 3 使之按倒相器 5 的输出 K 比特指定的移位比特数进行移位, 并将鼓式移位器 3 输出的 MSB 移位输入移位寄存器 4 内。

[将选择的路径选择信号寄存在 MSB]

[移位输入后上位 ($k-1$) 比特表示前一刻的状态]

[同时, 上位 ($k-1$) 比特的倒相成为用于选择下一个路径选择信号的移位比特数基础]

(3) 每进行一次 2^{k-1} 比特译码, 就把移位寄存器4的内容向数据存储器1中存储一次。

[选择的路径选择信号成为译码数据]

如上所述, 按照本发明, 在步骤(2)可以进行路径选择信号的选择和前一时刻的状态计算, 所以, 可以用 $(n+(n/2^{k-1})+1)$ 步进行n比特信息比的维特比译码中的跟踪反馈操作。

图5和图6是本发明其它实施例的运算装置的结构框图。图5和图6是在图4所示的结构上再加上第一寄存器7、ALU8、第1锁存器9、第2锁存器10、第2寄存器11, 12和第2总线13, 为了进行维特比译码的跟踪反馈操作以外的处理, 这些结构是必须的。

运算位宽度小于状态数 2^{k-1} 时, 或者末位不是终端, 连续使用卷积码时, 同样也可以进行跟踪反馈操作。

图7是本发明的一个实施例中运算装置的结构框图。在图7中, 21和22是存储路径量度和路径选择信号等的第1数据存储器 and 第2数据存储器, 23, 24分别是和数据存储器21, 22连接的用于进行数据的供给和运算结果的存储等的总线。25是用来选择来自总线24的输入和来自后面所述的寄存器的输入的多路转换器。26是暂时存储ALU8的右侧输入值的锁存器, 27是暂时存储ALU8的左侧输入值的锁存器。28是暂时存储加法器31的右侧输入值的锁存器, 29是暂时存储加法器31的左侧输入值的锁存器, 8是对锁存储器26和27的内容进行算术逻辑运算的ALU, 31是将锁存器28和29的内容进行加法运算的加法器, 32、33、34和35是用来暂时存储ALU8和加法器31的运算结

果的多个成对寄存器，36和37是用来选择寄存器32-35的输出的多路转换器，38是用来暂时存储ALU8的运算结果的第1寄存器，39是用来暂时存储加法器31的运算结果的第2寄存器，40是用来选择寄存器38和39的输出的多路转换器。41是大小比较器，用来对ALU8的输出和加法器31的输出进行大小比较，当ALU8的输出小或等于加法器31的输出时，输出比较结果0，当加法器31的输出小时，输出比较结果1。42是大小比较器41的比较结果即选择信号，43是将选择信号42进行移位输入的移位寄存器，44、45和46是表示数据存储器21、22的读取地址或存储地址的指示器，47是选择指示器输出的多路转换器，48是用来使选择信号42延迟1个步长的延迟器。

下面，对于上述结构的运算装置，对某时刻的 2^{k-1} 个状态的ACS计算动作进行说明。为了简单起见，取 $N=2$ ， $K=3$ 。预先在数据存储器21和22中，设定存储前一时刻各状态的路径量度 P^i ($i=0\sim 3$)，现时刻得到的各状态的路径量度 $P[i]$ ($i=0\sim 3$)和选择信号 $PS[i]$ ($i=0\sim 3$)的区域。图8是数据存储器21和22的区域分配的例子。在指示器44、45和46中，作为初始值分别预先设置(P^0 , P^2)的读取地址、

(P^0, P^2)的存储地址和选择信号的存储地址。图9对所有状态表示从前一时刻的状态向现时刻的状态转移的情况。在图9中，给出了从前一时刻的状态向现时刻的状态转移时，分别通过各卷积编码器的原来的输出代码的例子。所计算的接收信号和各个输出代码的距离为支路量度，只要输出代码相同，分路量度就相等，所以，可以预先计算输出代码00，01，10，11和 2^2 种支路量

度，并将已计算了 2^2 种的支路量度预先存储到寄存器 32、33、34 和 35 内。

下面，分步骤说明某时刻 2^2 个状态的 ACS 计算的动作概要。

状态 S[0] 的 ACS 计算

(1) 从由指示器 44 指示的数据存储器 21 和 22 的地址读取路径量度 $P^*[0]$ 、 $P^*[2]$ ，分别存储到锁存器 26 和锁存器 29 内。将寄存器 32 的内容存储到锁存器 27 内，将与之配对的寄存器 35 的内容存储到锁存器 28 内。ALU8 将锁存器 26 的内容和锁存器 27 的内容进行加法计算，并将其结果存储到寄存器 38 内，同时作为大小比较器 41 的一侧输入。另一方面，加法器 31 将锁存器 28 的内容和锁存器 29 的内容进行加法运算，并将其结果存储到寄存器 39 内，同时作为大小比较器 41 的另一侧输入。大小比较器 41 将 ALU8 的输出和加法器 31 的输出进行大小比较，产生选择信号 42。将选择信号 42 锁存到移位寄存器 43 和延迟器 48 内。该状态就是“求 $P^*[0]$ 和相对于输出代码 00 的支路量度之和，求 $P^*[2]$ 和相对于输出代码 11 的支路量度之和，并进行路径量度的比较、选择和路径选择信号的存储”。

(2) 将由延迟器 48 的输出所选择的寄存器 38 或寄存器 39 的内容存储到指示器 45 指示的数据存储器 21 的地址中。

状态 S[1] 的 ACS 计算

(3) 将寄存器 35 的内容存储到锁存器 27 内，将与之配对的寄存器 32 的内容存储到锁存器 28 内。ALU8 将在步骤 (1) 已存储到锁存器 26 内的内容和锁存器 27 的内容进行加法运算，并将其结果存储到寄存器 38 内，同时作为大小比较器 41 的一边

输入。另一方面，加法器 31 将在步骤 (1) 已存储到锁存器 29 内的内容和锁存器 28 的内容进行加法运算，并将其结果存储到寄存器 39 内，同时作为大小比较器 41 的另一边输入。大小比较器 41 将 ALU8 的输出和加法器 31 的输出进行比较，产生路径选择信号 42。将路径选择信号 42 锁存在移位寄存器 43 和延迟器 48 中。这一状态就是“ P^0 和相对于输出代码 11 的支路量度之和，求 P^2 和相对于输出代码 00 的支路量度之和，并进行路径量度的比较、选择和路径选择信号的存储”。

(4) 使指示器 45 增值后，将由延迟器 48 的输出所选择的寄存器 38 或寄存器 39 的内容存储到指示器 45 指示的数据存储器 21 的地址中。

(5) 使指示器 45 的内容返回到 (P^0, P^2) 的存储地址。

状态 S^2 的 ACS 计算

(6) 使指示器 44 增值后，从指示器 44 指示的数据存储器 21 和 22 的地址读取路径量度 P^1 ， P^3 ，分别将它们存储到锁存器 26 和锁存器 27 内。将寄存器 33 的内容存储到锁存器 27 内，将与之配对的寄存器 34 的内容存储到锁存器 28 内。ALU8 将锁存器 26 的内容和锁存器 27 的内容进行加法运算，并将其结果存储到寄存器 38 内，同时，作为大小比较器 41 的一边输入。另一方面，加法器 31 将锁存器 28 的内容和锁存器 29 的内容进行加法运算，并将其结果存储到寄存器 39 内，同时作为大小比较器 41 的另一边输入。大小比较器 41 将 ALU8 的输出和加法器 31 的输出进行大小比较，产生路径选择信号 42。将路径选择信号 42 锁存到移位寄存器 43 和延迟器 48 内。这一状态就是“求 P^1

和相对于输出代码 01 的支路量度之和，求 $P^{[3]}$ 和相对于输出代码 01 的支路量度之和，并进行路径量度的比较、选择和路径选择信号的存储”。

(7) 将由延迟器 48 的输出所选择的寄存器 38 或寄存器 39 的内容存储到指示器 45 指示的数据存储器 22 的地址中。

状态 $S^{[3]}$ 的 ACS 计算

(8) 将寄存器 34 的内容存储到锁存器 27 内，将与之配对的寄存器 33 的内容存储到锁存器 28 内。ALU8 将在步骤 (6) 已存储到锁存器 26 内的内容和锁存器 27 内的内容进行加法运算，并将其结果存储到寄存器 38 内，同时作为大小比较器 41 的一边输入。另一方面，加法器 31 将在步骤 (6) 已存储到锁存器 28 内的内容和锁存器 29 的内容进行加法运算，并将其结果存储到寄存器 39 内，同时作为大小比较器 41 的另一边输入。大小比较器 41 将 ALU8 的输出和加法器 31 的输出进行大小比较，产生路径选择信号 42。将路径选择信号 42 锁存到寄存器 43 和延迟器 48 内。这一状态就是“求 $P^{[1]}$ 和相对于输出代码 10 的支路量度之和，求 $P^{[3]}$ 和相对于输出代码 01 的支路量度之和，并进行路径量度的比较、选择和路径选择信号的存储”。

(9) 使指示器 45 增值后，将由延迟器 48 的输出所选择的寄存器 38 或寄存器 39 内的内容存储到指示器 45 指示的数据存储器 22 的地址中。

(10) 将移位寄存器 43 内的内容存储到指示器 46 指示的数据存储器 21 的地址中。这一状态就是“将前一时刻各状态的路径选择信号填在 1 个字内进行存储”。

这样，在上述实施例的运算装置中，对于 $N = 2$ ， $K = 3$ 的情况，为了进行某一时刻 2^2 个状态的 ACS 计算，相对于以往的大约 37 个步骤，用本发明的装置大约 10 个步骤就可以完成。另外，通常，对于 $N = 2$ ， $K = 3$ 的情况，为了进行某时刻的 2^{k-1} 个状态的 ACS 计算，相对于以往的大约 $(9 \cdot 2^{k-1} + 1)$ 个步骤，用大约 $(2 \cdot (2^{k-1} + 1))$ 个步骤就能完成。只是当 N 增大或寄存器数量有限制时，多少要加一些步骤。

下面，参照附图详细说明本发明的运算装置的实施例。

图 10 是本发明运算装置的实施例的结构框图。在图 10 中，该运算装置具有数据存储器 51 及 52、总线 53 及 54 和多路转换器 55，数据存储器 51 及 52 用来存储接收信号和距离等；总线 53 及 54 和数据存储器 51 及 52 连接，用来进行数据的供给及运算结果的存储等；多路转换器 55 用来选择来自总线 54 的输入和来自多路转换器 66 的输入，并且该运算装置还具有锁存器 56、57、58 和 59，锁存器 56 用来暂时存储算术逻辑运算电路（以下简称为 ALU）8 的右侧输入值；锁存器 57 用来暂时存储 ALU8 的左侧输入值；锁存器 58 用来暂时存储加法器 31 的右侧输入值；锁存器 59 用来暂时存储加法器 31 的左侧输入值。另外，该运算装置还具有进行算术逻辑运算的 ALU8、加法器 31、暂时存储运算结果的寄存器 62，63，64 及 65、选择这些寄存器 62，63，64，65 的输出的多路转换器 66 及 67 和指示数据存储器 51 及 52 的读取地址或存储地址的指示器 68。

下面，说明本实施例的构成的动作。

这里，先说明求某时刻 2^N 种支路量度的动作。只要是有接收

信号和现实性的规模，对于接收信号 R 取得的所有值，将距离 $d(R,0)$ 作为预先求出的表存储到数据存储器 51 内。对于接收信号 R 取得的所有值，将距离 $d(R,1)$ 作为预先求出的表存储到数据存储器 52 内。在数据存储器 51 和 52 内， $d(R,0)$ 和 $d(R,1)$ 存储在同一地址。另外，在数据存储器 51 中存储的接收信号是为了查表用的地址。同时使用 ALU8 和加法器 31 时，将寄存器 62 与寄存器 65 及寄存器 63 与寄存器 64 分别配对使用。

下面，对于 $N = 2$ 的情况，按处理顺序说明求支路量度的动作概要。

(1) 从数据存储器 51 读取接收信号 R_0 ，存储到指示器 68 内。

(2) 从指示器 68 指示的数据存储器 51 的地址读取距离 $d(R_0,0)$ ，存储到锁存器 56 内。ALU8 使锁存器 56 的内容通过后，存储到寄存器 62 内。另一方面，从指示器 68 指示的数据存储器 52 的地址读取距离 $d(R_0,1)$ ，存储到锁存器 59 内。加法器 31 使锁存器 59 的内容通过后，存储到寄存器 65 内。

(3) ALU8 使锁存器 56 的经过上述 (2) 的处理所存储的内容通过后存储到寄存器 63 内。另一方面，加法器 31 使锁存器 59 的经过上述 (2) 的处理所存储的内容通过后存储到寄存器 64 内。

(4) 从数据存储器 51 读取接收信号 R_1 ，将其存储到指示器 68 内。

(5) 从指示器 68 指示的数据存储器 51 的地址读取距离 $d(R_1,0)$ 后，存储到锁存器 56 内。将寄存器 62 的内容存储到锁存

器 57 内。ALU8 将锁存器 56 的内容和锁存器 57 的内容进行加法运算，并将其结果存储到寄存器 62 内。另一方面，从指示器 68 指示的数据存储器 52 的地址读取距离 $d(R_1,1)$ ，存储到锁存器 59 内。将寄存器 65 的内容存储到锁存器 59 内。加法器 31 将锁存器 58 的内容和锁存器 59 的内容进行加法运算，并将其结果存储到寄存器 65 内。

因此，在寄存器 62 内可以求出 R_0 ， R_1 和 00 的距离，在寄存器 65 内可以求出 R_0 ， R_1 和 11 的距离。

(6) 将寄存器 64 的内容存储到锁存器 57 内。ALU8 将锁存器 56 的经过上述 (5) 的处理后存储的内容和锁存器 57 的内容进行加法运算，并将其结果存储到寄存器 64 内。另一方面，将锁存器 63 的内容存储到锁存器 58 内。加法器 31 将锁存器 59 的经过上述 (5) 的处理后存储的内容和锁存器 58 的内容进行加法运算，并将其结果存储到寄存器 63 内。因此，在寄存器 63 内可以求出 R_0, R_1 和 01 的距离，在寄存器 64 内可以求出 R_0 ， R_1 和 10 的距离。

如上所述，按照本实施例，对于 $N = 2$ 的情况，求 2^2 种支路量度时的处理用大约 6 步就可以完成。通常，对于 $N = N$ 的情况，为了求 2^N 种支路量度，进行 $N (2^N/2+1)$ 步处理就可以完成。只是当 N 增大，寄存器的数量有限制时，多少要增加若干处理步骤。另外，所具备的表不现实时，需要根据接收信号求出每次的距离 $d(R,0)$ 、 $d(R,1)$ ，增加这部分所需要的处理步骤。即，和通常的处理一样。

从上述说明可知，由于本发明的运算装置可以利用 ALU 和加

法器同时进行加法运算，所以，可以同时计算两个支路量度。这时，把数据存储器分为两部分，不增加存储器的大小，就可以减少进行支路量度计算的运算步骤数。

说明书附图

图 1

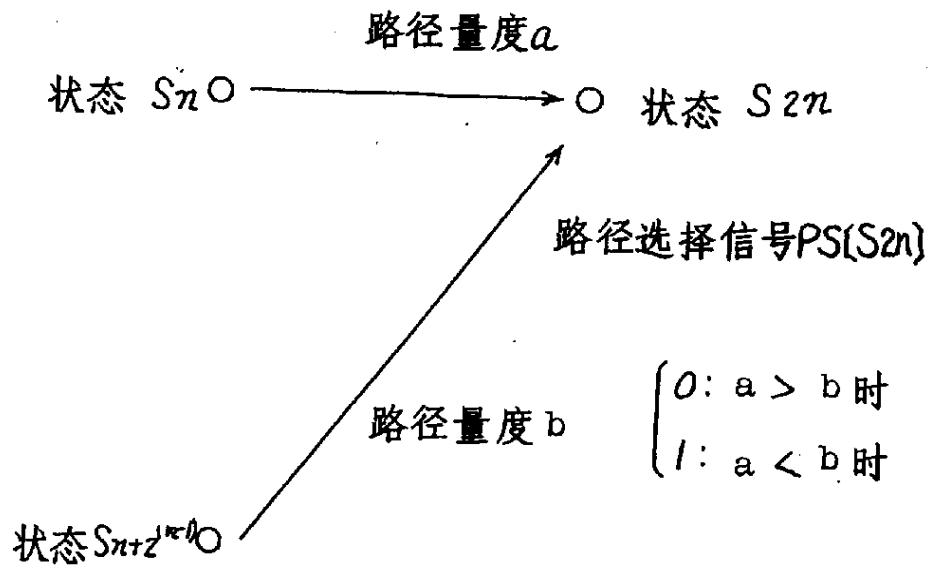
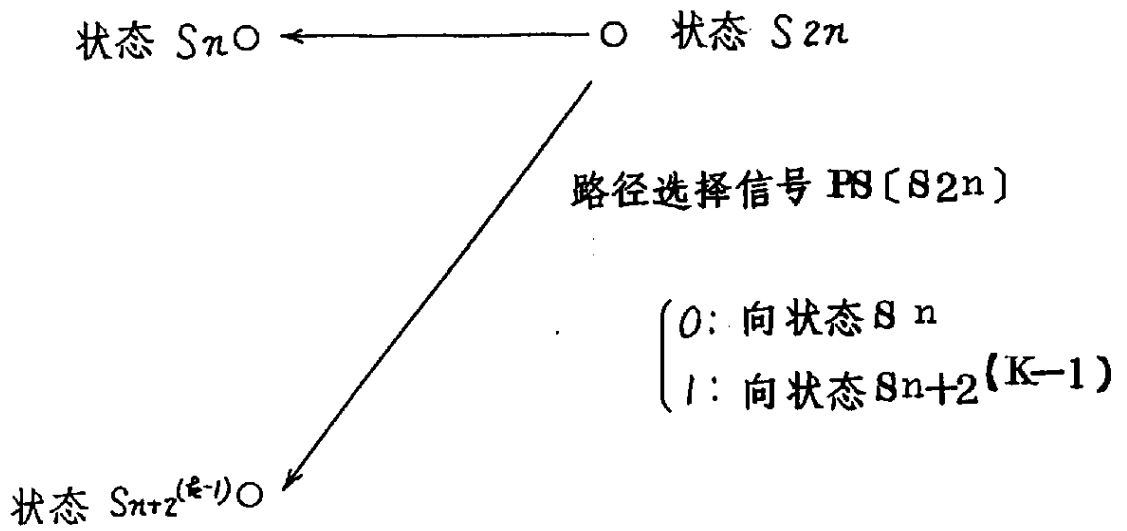


图 2



前一时刻的状态 $S_i + PS[S_i] \cdot 2^{(k-1)}$

图 3

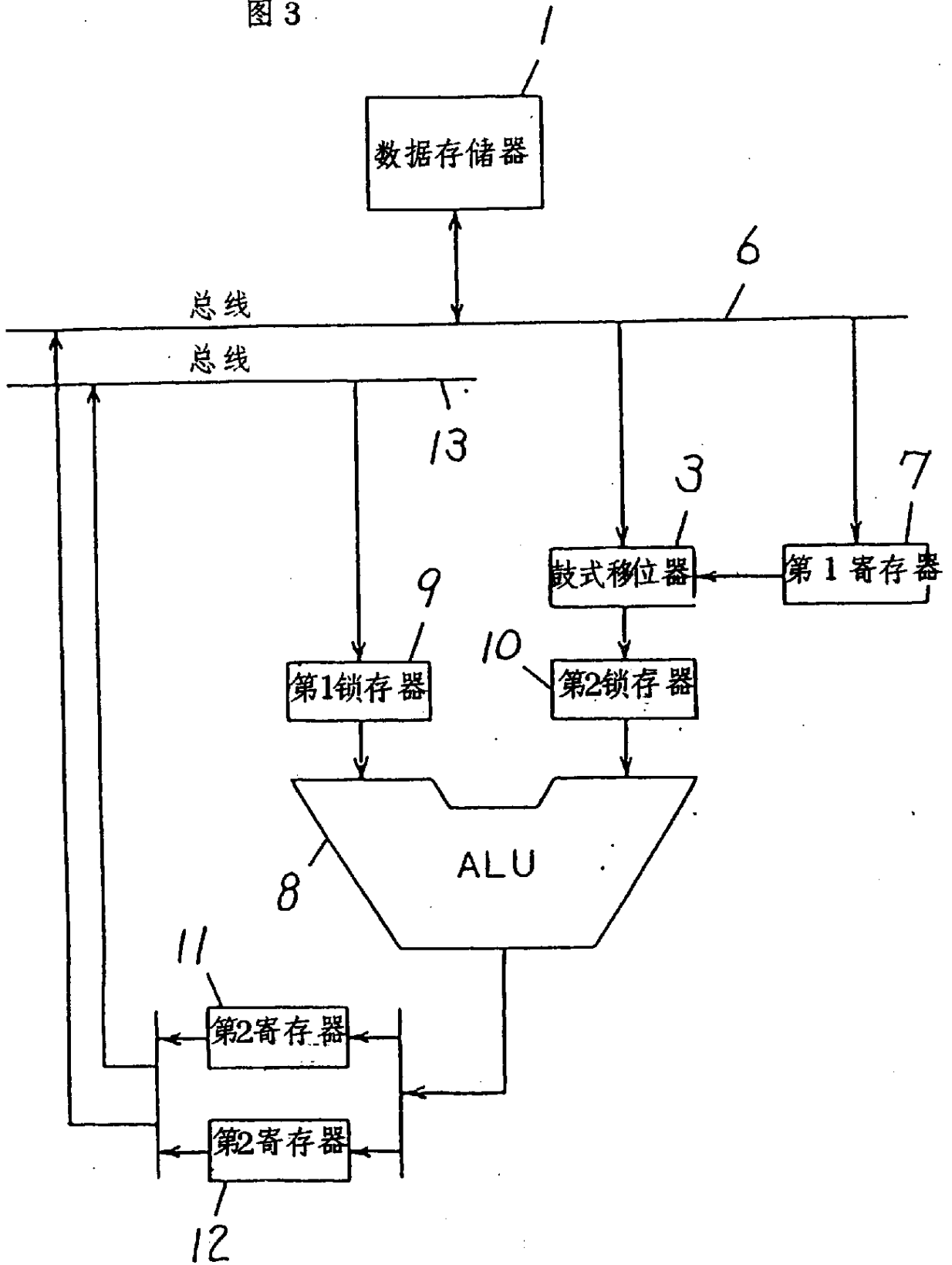


图 4

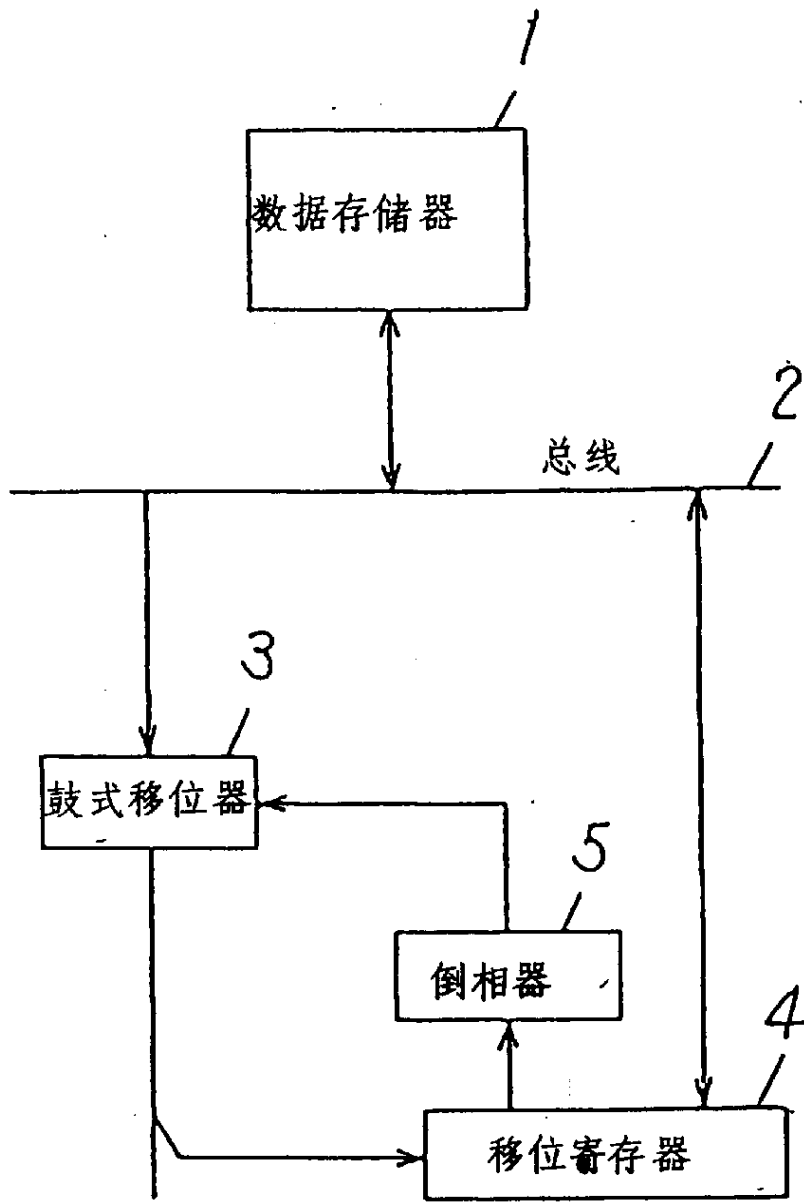


图 5

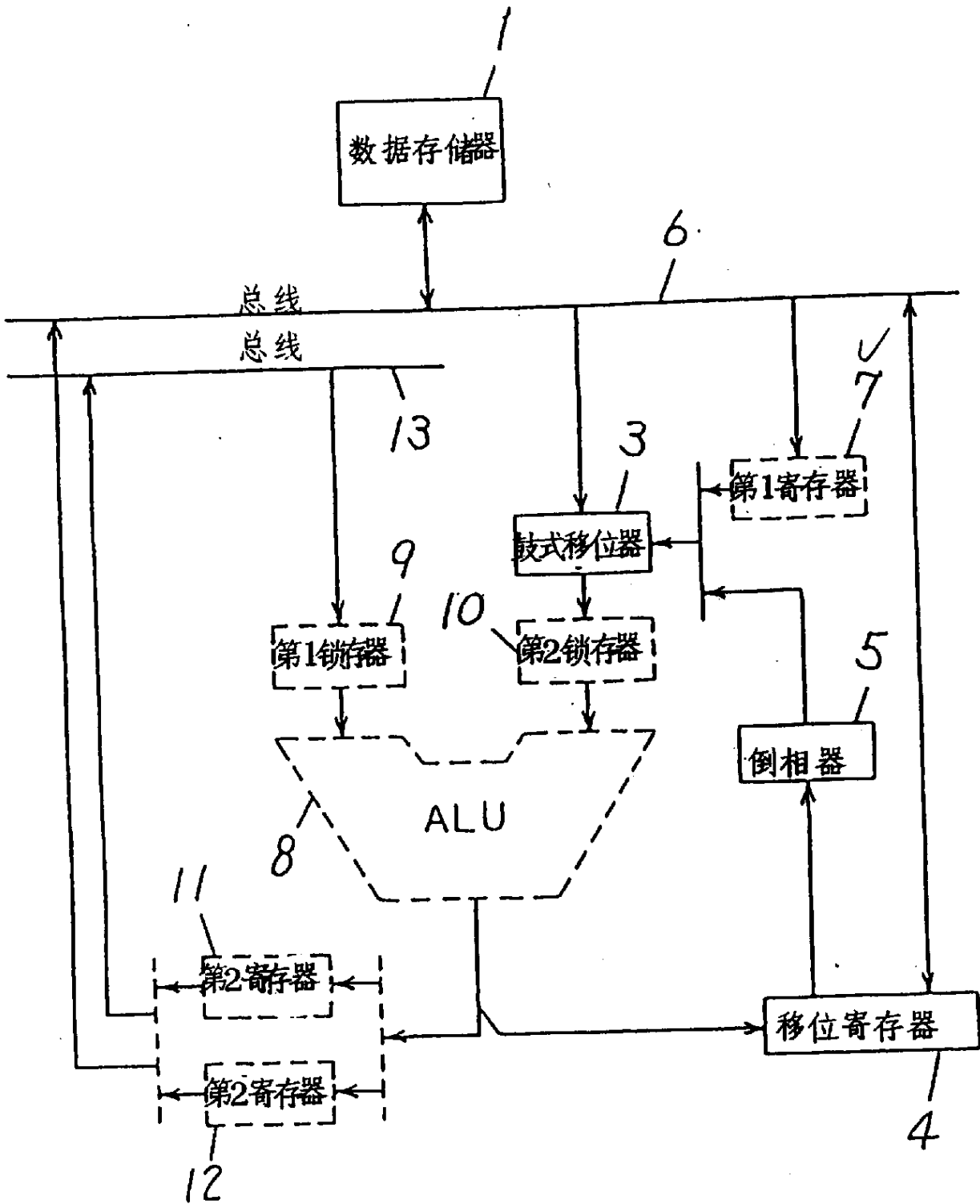


图 6

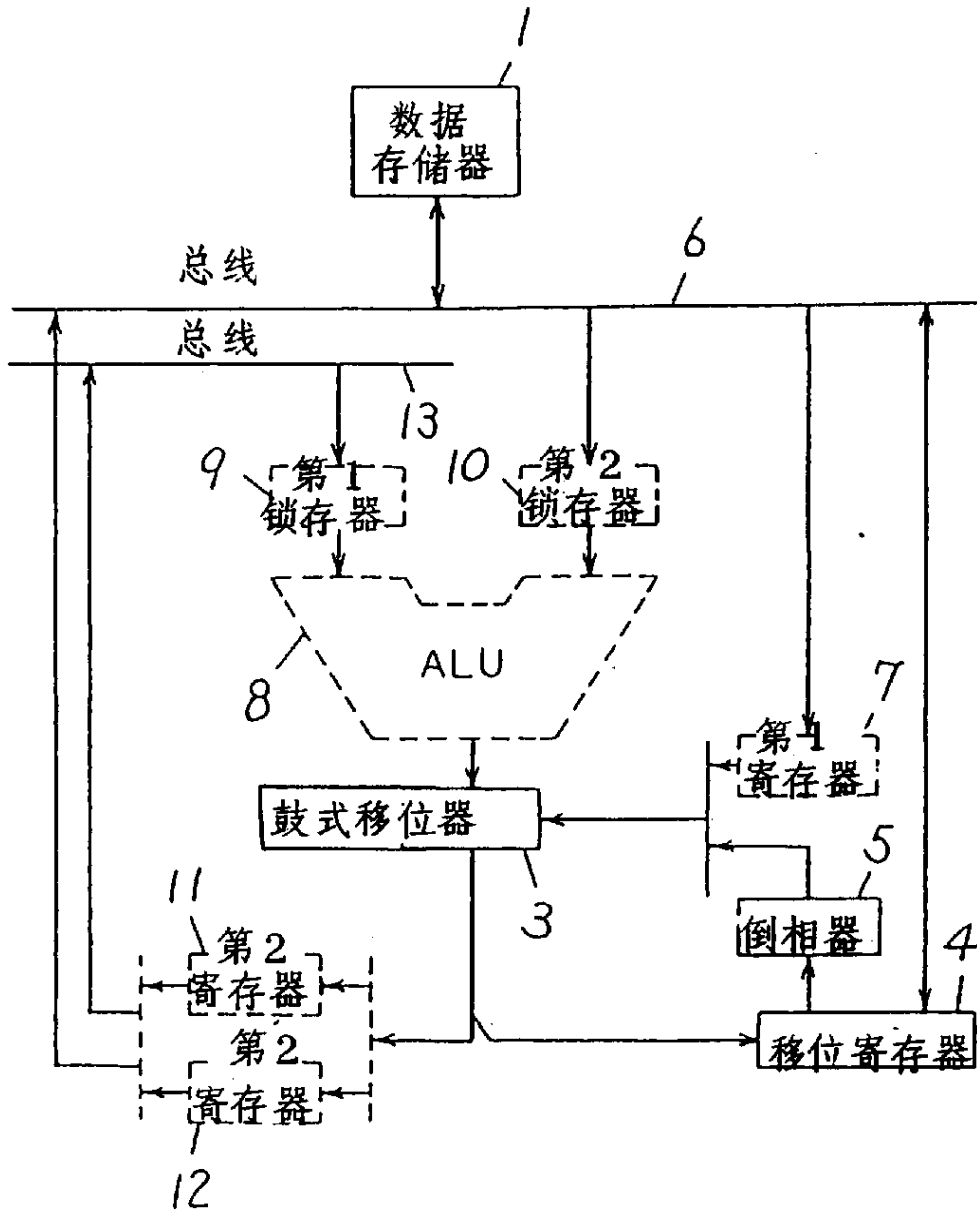


图7

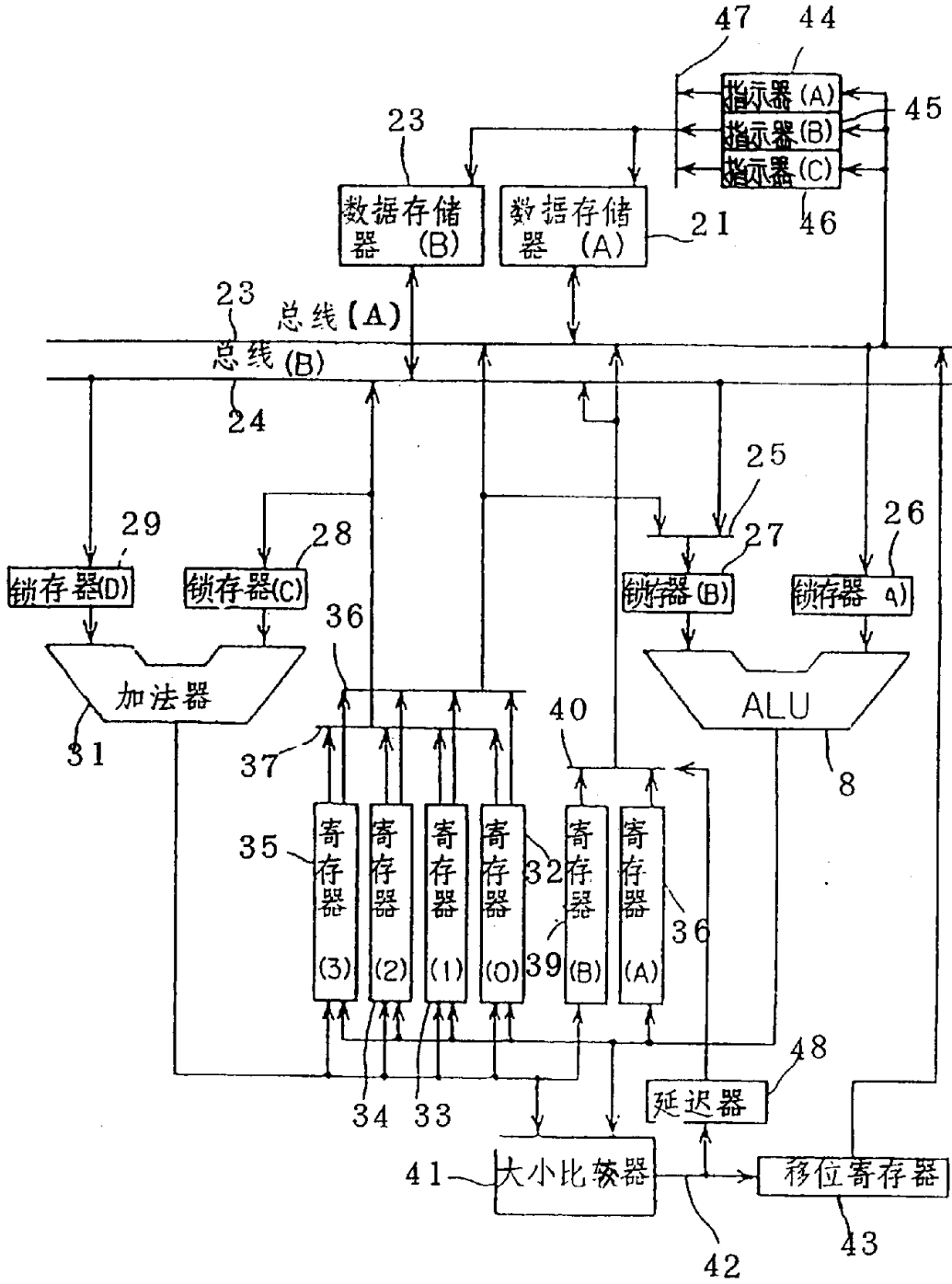


图 8

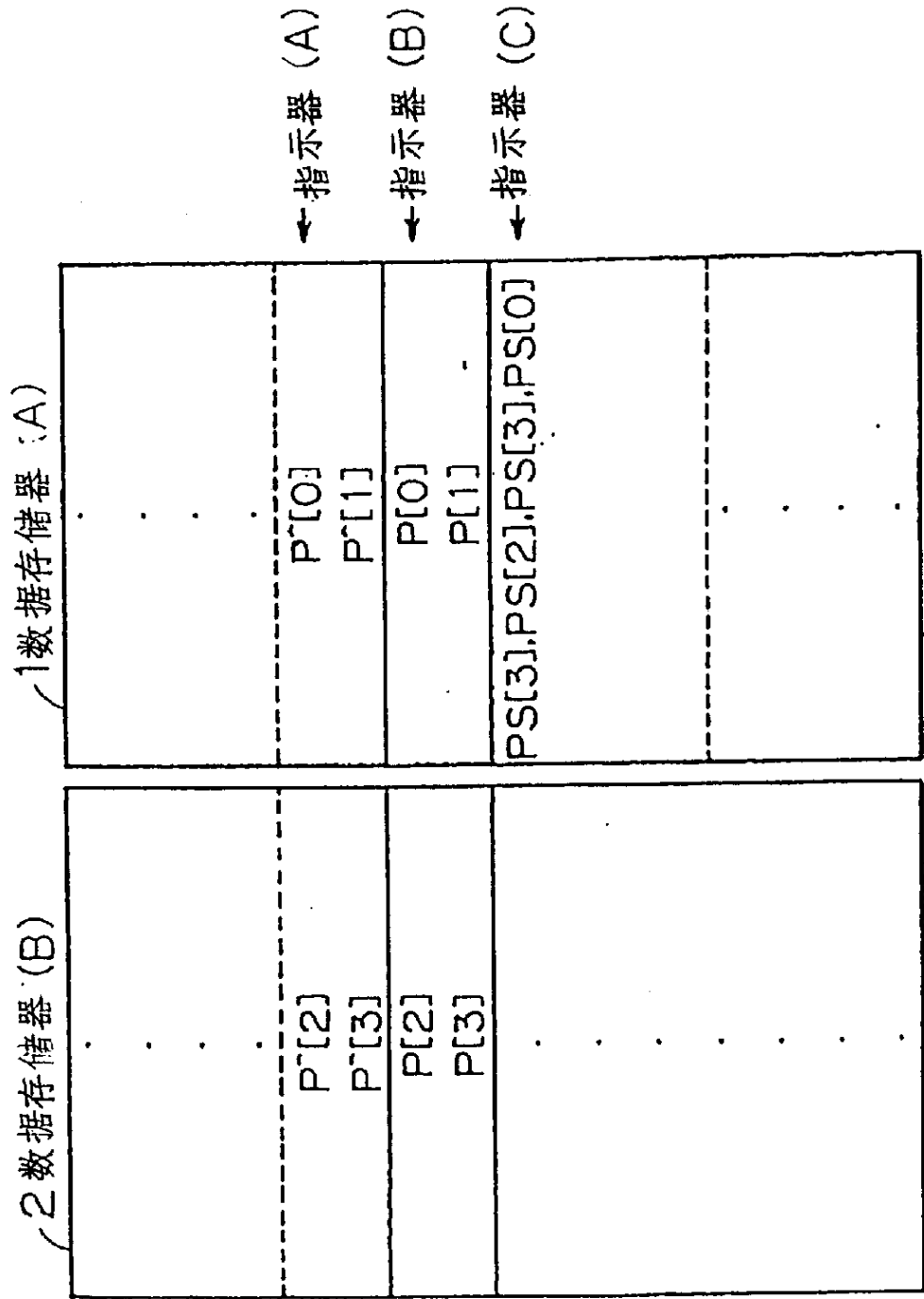


图 9

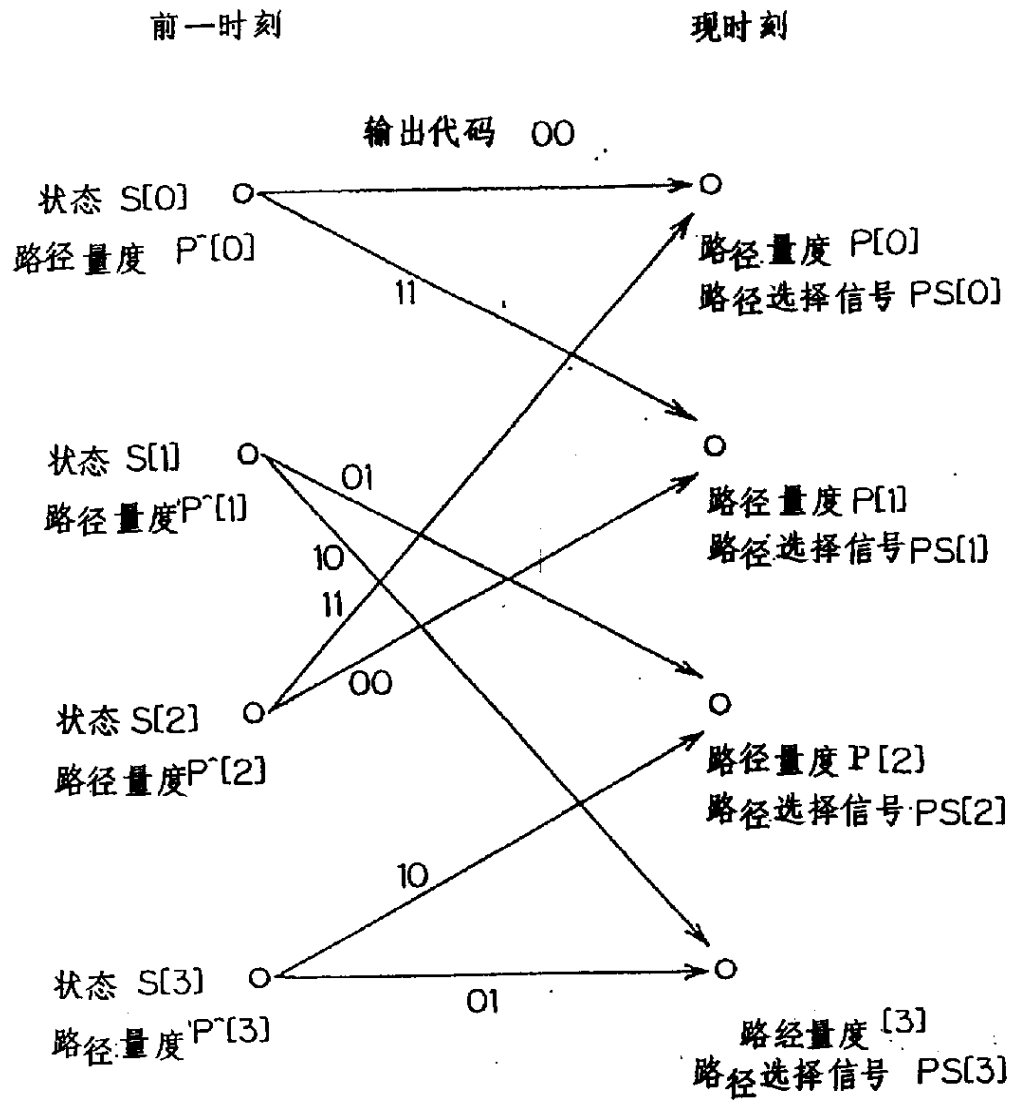


图 10

