

[19] 中华人民共和国国家知识产权局

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

H04L 1/00 H04L 1/20

H04L 1/08 H04L 25/02

H04L 25/03 H03M 13/41

## [12] 发明专利说明书

[21] ZL 专利号 94193504.3

[45] 授权公告日 2002 年 12 月 11 日

[11] 授权公告号 CN 1096163C

[22] 申请日 1994.9.23 [21] 申请号 94193504.3

[74] 专利代理机构 上海专利商标事务所

[30] 优先权

代理人 陈亮

[32] 1993.9.24 [33] US [31] 126,477

[86] 国际申请 PCT/US94/10774 1994.9.23

[87] 国际公布 WO95/08888 英 1995.3.30

[85] 进入国家阶段日期 1996.3.25

[73] 专利权人 夸尔柯姆股份有限公司

地址 美国加利福尼亚州

[72] 发明人 丹尼尔 R · 金得里得

布赖恩 K · 巴特勒

伊弗雷姆 · 泽哈维 杰克 K · 沃乐夫

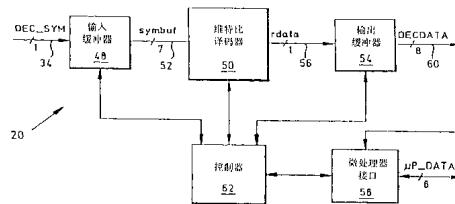
审查员 焦景梅

权利要求书 12 页 说明书 30 页 附图 12 页

[54] 发明名称 码分多址系统中提供译码器位数据的方法及多速率译码器

[57] 摘要

本发明涉及一种在码分多址(CDMA)移动通信系统(22)中恢复被卷积编码成码符号流的原始比特数据流的维特比译码器(20)。译码器(20)同时以几种与某些多速率译码器相关联的数据速率进行译码。译码器(20)能以连续模式或划分成帧的模式以未知的数据速率进行译码。这是通过同时以多种速率进行译码，并产生一个或一个以上的每组经译码数据分组的数据质量量度来实现的。



1. 一种响应于表示数据速率为  $R_i$  的原始位数据的码符号数据的传输用以提供经译码的位数据的方法，其中， $R_i$  为两种或更多种预定的原始位数据速率中的一种数据速率，该方法用包括输入缓冲器、卷积译码器和输出缓冲器的译码器装置来执行，该方法包括下列步骤：

以预定持续时间的帧形式传输所述码符号数据，其中使所述码符号数据的每个所述顺序帧之间的转换强制进入预定的状态；

在所述输入缓冲器内接收并存储所述码符号数据的所述传输的顺序段；

在所述卷积译码器中把所述顺序段中的任一段译码成两组或两组以上的经译码的位数据分组  $\{P_i\}$ ，每个分组  $P_i$  包括：

对应于所述数据速率为  $R_i$  的原始位数据的经译码的位数据  $I_i$ ；和

表示所述任一段内的符号差错情况和所述经译码的位数据内的数据差错情况的质量量度数据  $Q_i$ ；以及

把所述两组或两组以上的分组存储在所述输出缓冲器内。

2. 如权利要求 1 所述的方法，其特征在于，表示各所述经译码的位数据内每一帧中的数据差错情况的所述质量量度数据  $Q_i$  包括：

质量量度 (QM)，表示预定的质量阈值 (QT) 与在所述经译码比特的数据的顺序帧之间进行每次所述转换时对所述零状态进行译码的概率大小的比较结果。

3. 如权利要求 1 所述的方法，其特征在于，所述码符号数据根据编码算法表示所述原始位数据，该方法进一步包括下列步骤：

根据所述编码算法对至少两组经译码的位数据分组  $\{P_i\}$  的每组进行重编码，以产生局部码符号数据分组  $\{L_i\}$ ；

把所述码符号数据与所述至少两组的局部码符号数据分组

(Li) 中的每组比较，以产生它们之间的差的质量量度 (Qi)；

把至少两个质量量度 {Qi} 存储在所述输出数据缓冲器内。

4. 如权利要求 1 所述的方法，其特征在于，具有第一位数据速率的数据帧包括经编码的数据比特分组，每个分组包括循环冗余检查比特，该方法包含另外的无顺序的步骤：

用所述循环冗余检查比特确定所述数据比特分组的差错率；

产生指示所述差错率的质量量度 (Qi)；

把至少两个质量量度 {Qi} 存储在所述输出缓冲器内。

5. 一种响应于表示数据速率为  $R_i$  的原始位数据的码符号数据的传输用以提供经译码的位数据的多速率译码器，其中， $R_i$  为两种或更多种预定的原始位数据速率中的一种数据速率，该多速率译码器包含：

输入缓冲器装置，接收并存储所述码符号数据的所述传输的顺序段；所述码符号数据是以预定持续时间的帧形式来传输，所述码符号数据的每个所述顺序帧之间的转换强制进入预定的状态；

译码器装置，连接到所述输入缓冲器装置，把所述顺序段中的任一段译码成两组或两组以上的经译码的位数据分组 {Pi}，每个分组  $P_i$  包括：

对应于所述数据速率为  $R_i$  的原始位数据的经译码的位数据  $I_{i1}$ ；和

表示所述任一段内的符号差错情况和所述经译码的位数据内的数据差错情况的质量量度数据  $Q_i$ ；以及

输出缓冲器装置，联接到所述译码器装置，存储所述两组或多组分组。

6. 如权利要求 5 所述的多速率译码器，其特征在于，所述编码符号数据表示经编码的原始位数据的一串  $N_i$  个复制品。

7. 如权利要求 6 所述的多速率译码器，其特征在于，进一步包含：

连接到所述译码器装置上的质量量度装置，产生表示预定的质

量阈值 (QT) 与在所述经译码的位数据的顺序帧之间进行每次所述转换时对所述零状态进行译码的概率大小的比较结果的 Yamamoto 质量量度 (YQM)。

8. 在一种通信系统中，根据第一编码算法表示原始位数据的码符号数据在一个或一个以上的信道内以第一组多个预定的原始位数据速率 {R<sub>i</sub>} 中的一种速率，或以第一模式连续不断地或以第二模式以预定持续时间的帧形式，进行传输，每个所述帧表示经编码的原始位数据的第一种形式和 (N<sub>i</sub>-1) 种重复形式，通过产生对应于所述原始位数据的所述第一种形式的经译码的位数据由所述码符号数据恢复所述原始位数据的方法包含下列无顺序的步骤：

(a) 在所述输入缓冲器内接收和存储至少一个码符号数据的所述帧；

(b) 对所述编码符号数据进行译码，以产生对应于所述第一组预定的原始位数据速率 {R<sub>i</sub>} 中至少两种速率中的每种的所述经译码的位数据的分组 (P<sub>i</sub>)；以及

(c) 把所述至少两组经译码的位数据分组 {P<sub>i</sub>} 存储在输出数据缓冲器内；

(d) 根据所述第一种编码算法对所述至少两组经译码的位数据分组 {P<sub>i</sub>} 中的每一组进行重编码，以产生局部码符号数据 (L)；

(e) 把所述码符号数据与所述至少两组局部码符号数据分组 {L<sub>i</sub>} 中的每组比较，产生它们之间的差的质量量度 (Q<sub>i</sub>)；

(f) 把所述至少两个质量量度 {Q<sub>i</sub>} 存储在所述输出数据缓冲器内。

9. 如权利要求 8 所述的方法，其特征在于，所述顺序译码步骤

(b) 包含下列无顺序步骤：

(b. 1) 根据第二种编码算法把至少一个符号量度值赋予每个所述码符号数据；

(b. 2) 根据选出的所述符号量度值中的值对应于每个所述码符

号基准指定表示原始数据比特转换概率的分支量度；

(b. 3) 根据最可能的前一所述状态量度值与从其得到的所述分支量度值之和对应于所述码符号基准指定表示原始数据比特转换概率的状态量度值；

(b. 4) 把所述状态量度值存储在路径存储器内；

(b. 5) 根据最可几的所述状态量度值，选择每个所述原始数据位的最可几值，该最可几所述状态量度值对应于在与所述路径存储器内的所述每个码符号相对应的最可几的所述状态量度值之前的判决路径链。

10. 如权利要求 9 所述的方法，其特征在于，按所述第一模式以所述原始位数据速率 {R<sub>i</sub>} 中的一种速率连续地发送所述码符号数据。

11. 如权利要求 10 所述的方法，其特征在于，所述码符号数据表示每个所述经编码的原始数据比特的一串 N<sub>i</sub> 个复制品。

12. 如权利要求 10 所述的方法，其特征在于，码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的(N<sub>i</sub>-1) 个复制品的序列。

13. 如权利要求 8 所述的方法，其特征在于，按所述第一模式以所述原始位数据速率 {R<sub>i</sub>} 中的一种速率连续地发送所述码符号数据。

14. 如权利要求 8 所述的方法，其特征在于所述码符号数据表示每个所述经编码的原始数据比特的一串 N<sub>i</sub> 个复制品。

15. 如权利要求 8 所述的方法，其特征在于，码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的(N<sub>i</sub>-1) 个复制品的序列。

16. 如权利要求 8 所述的方法，其特征在于，按所述第一模式以所述原始位数据速率 {R<sub>i</sub>} 中的一种速率连续地发送所述码符号数据。

17. 如权利要求 8 所述的方法，其特征在于，所述译码步骤(b)

包含：

(b. 1) 根据第二种编码算法把至少一个符号量度值赋予每个所述码符号数据；

(b. 2) 根据选出的所述符号量度值中的值对应于每个所述编码符号基准指定表示原始数据比特转换概率的分支量度值；

(b. 3) 根据最可能的前一所述状态量度值与从其得到的所述分支量度值之和对应于所述编码符号基准指定表示原始数据比特转换概率的状态量度值；

(b. 4) 把所述状态量度值存储在路径存储器内；

(b. 5) 根据最可几的所述状态量度值，选择每个所述原始数据位的最可能的值，该最可几所述状态量度值对应于在与所述路径存储器内的所述每个码符号相对应的最可几的所述状态量度值之前的判决路径链。

18. 如权利要求 8 所述的方法，其特征在于，所述码符号数据表示每个所述经编码的原始数据比特的一串  $N_i$  个复制品。

19. 如权利要求 8 所述的方法，其特征在于，码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的  $(N_i-1)$  个复制品的序列。

20. 在一种通信系统中，根据第一编码算法表示原始位数据的码符号数据在一个或一个以上的信道内以第一组多个预定的原始位数据速率  $\{R_i\}$  中的一种速率，或以第一模式连续不断地或以第二模式以预定持续时间的帧形式进行传输，每个所述帧表示编码的原始位数据的第一种形式和重复形式，其中由多速率译码器在输出端上产生对应于所述经编码的原始位数据的所述第一种形式的经译码的位数据，所述多速率译码器包含：

输入缓冲器装置，至少接收并存储所述码符号数据的一个所述帧；

译码器装置，与所述输入缓冲器装置相连，根据所述码符号数据产生对应于至少两种所述预定的原始位数据速率  $\{R_i\}$  中的每

种的经译码的位数据分组 ( $P_i$ )；

输出缓冲器装置，与所述顺序译码器装置相连，存储所述至少两组所述经译码的位数据分组；

在所述输入缓冲器装置内具有一符号传送装置，为至少两种的所述预定原始位数据速率  $\{R_i\}$  中的每一种速率选择对应于所述经编码的原始位数据的所述第一种形式的一组所述码符号数据 ( $S_i$ )，并把所述组 ( $S_i$ ) 传送到所述顺序译码器装置。

21. 如权利要求 20 所述的多速率译码器，其特征在于，进一步包含：

在所述译码器装置内的质量量度装置，产生每个所述经译码的位数据分组 ( $P_i$ ) 的质量量度 ( $Q_i$ )，所述质量量度 ( $Q_i$ ) 表示与所述每个经译码的数据比特分组 ( $P_i$ ) 相关联的码符号数据差错的数值。

22. 如权利要求 21 所述的多速率译码器，其特征在于，所述质量量度装置包含：

数据重编码装置，根据所述第一编码算法对所述至少两个经译码的位数据分组  $\{P_i\}$  中的每个分组进行重编码，产生局部码符号数据分组 ( $L_i$ )；和

比较装置，与所述数据重编码装置相连，把每个所述局部码符号数据分组 ( $L_i$ ) 与每个所述码符号数据组 ( $S_i$ ) 比较，并对它们之间的差进行计数。

23. 如权利要求 22 所述的多速率译码器，其特征在于，所述译码器装置包含：

符号量度装置，根据第二编码算法把符号量度值赋予每个所述码符号；

分支量度计算装置，与所述符号量度装置相连，根据相应的所述符号量度值产生一对分支量度值，它表示对应于来自所述输入缓冲器装置的所述每个码符号的原始数据比特转换的加权概率；

判决路径装置，与所述分支量度计算装置相连，根据相应的所

述分支量度值产生并存储对应于来自所述输入缓冲器装置的所述每个码符号的每次可能的原始数据比特转换的状态量度判决值；和

路径反向链接装置，与所述判决路径装置相连，对每个所述原始数据比特转换选择最可几判决路径，并产生相应的所述经译码的数据比特。

24. 如权利要求 23 所述的多速率译码器，其特征在于，是以一块单片集成电路的形式来实现。

25. 如权利要求 24 所述的多速率译码器，其特征在于，所述码符号数据表示每个所述经编码的原始数据比特的一串  $N_i$  个复制品。

26. 如权利要求 24 所述的多速率译码器，其特征在于，所述码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的  $(N_i-1)$  个复制品的序列。

27. 如权利要求 20 所述的多速率译码器，其特征在于，进一步包含：

在所述译码器装置内的质量量度装置，产生每个所述经译码的位数据分组  $(P_i)$  的质量量度  $(Q_i)$ ，所述质量量度  $(Q_i)$  符号数据误差与所述每个译码数据比特分组  $(P_i)$  相关联。

28. 如权利要求 27 所述的多速率译码器，其特征在于，所述质量量度装置包含：

数据重编码装置，根据所述第一编码算法对所述至少两个经译码的位数据分组  $\{P_i\}$  中的每个分组进行重编码，产生局部码符号数据分组  $(L_i)$ ；和

比较装置，与所述数据重编码装置相连，把每个所述局部码符号数据分组  $(L_i)$  与每个所述码符号数据组  $(S_i)$  比较，并对它们之间的差进行计数。

29. 如权利要求 28 所述的多速率译码器，其特征在于，所述码符号数据表示每个所述原始数据比特的一串  $N_i$  个复制品。

30. 如权利要求 28 所述的多速率译码器，其特征在于，所述码符号数据表示完整的所述第一原始位数据形式加上所述完整的第一原始位数据形式的( $N_i-1$ )个复制品的序列。

31. 如权利要求 20 所述的多速率译码器，其特征在于，所述码符号数据表示每个所述原始数据比特的一串  $N_i$  个复制品。

32. 如权利要求 20 所述的多速率译码器，其特征在于，码符号数据表示完整的所述第一原始位数据形式加上所述完整的第一原始位数据形式的( $N_i-1$ )个复制品的序列。

33. 如权利要求 20 所述的多速率译码器，其特征在于，所述译码器装置包含：

    符号量度装置，根据第二编码算法把符号量度值赋予每个所述码符号；

    分支量度计算装置，与所述符号量度装置相连，根据相应的所述符号量度值产生一对分支量度值，它表示对应于来自所述输入缓冲器装置的所述每个码符号的原始数据比特转换的加权概率；

    判决路径装置，与所述分支量度计算装置相连，根据相应的所述分支量度值产生并存储对应于来自所述输入缓冲器装置的所述每个码符号的每次可能的原始数据比特转换的状态量度判决值；  
    和

    路径反向链接装置，与所述判决路径装置相连，对每个所述原始数据比特转换选择最可几判决路径，并产生相应的所述经译码的数据比特。

34. 如权利要求 20 所述的多速率译码器，其特征在于，基本上以一块单片集成电路的形式来实现。

35. 一种多速率译码器，对在输入端接收到的码符号数据进行译码，在输出端得到经译码的位数据，所述码符号数据根据第一编码算法表示原始位数据，以第一组多个预定的原始位数据速率 { $R_i$ } 中的一种速率，或按第一模式连续不断地或按第二模式以预定持续时间的帧形式，传输所述码符号数据，每个所述帧表示

经编码的原始位数据的第一种形式和( $N_i - 1$ )种重复形式，其中  $N_i$  和  $i$  为非零的正整数，所述多速率译码器包含：

输入缓冲器装置，连接到所述输入端，至少接收并存储所述码符号数据的一个所述帧；

译码器装置，与所述输入缓冲器装置相连，根据所述码符号数据产生对应于至少两种所述预定的原始位数据速率  $\{R_i\}$  中的每种的经译码的位数据分组  $(P_i)$ ；

输出缓冲器装置，与所述顺序译码器装置相连，存储所述至少两组所述译码的位数据分组；

第一选择装置，与所述输入缓冲器装置相连，选择每种都对应于不同的所述第一编码算法的多种译码模式中的一种；和

第二选择装置，与所述输入缓冲器装置相连，选择对应于连续的和划分为帧形式的原始位数据的所述第一和第二信道模式中的一种。

36. 如权利要求 35 所述的多速率译码器，其特征在于，进一步包含：

在所述输入缓冲器装置内的符号传送装置，为至少两种的所述预定原始位数据速率  $\{R_i\}$  中的每一种速率选择对应于所述原始位数据的所述第一种形式的一组所述码符号数据  $(S_i)$ ，并把所述组  $(S_i)$  传送到所述顺序译码器装置。

37. 如权利要求 36 所述的多速率译码器，其特征在于，进一步包含：

在所述译码器装置内的质量量度装置，产生每个所述经译码的位数据分组  $(P_i)$  的质量量度  $(Q_i)$ ，所述质量量度  $(Q_i)$  符号数据差错与所述每个经译码的数据比特分组  $(P_i)$  相关联。

38. 如权利要求 37 所述的多速率译码器，其特征在于，所述质量量度装置包含：

数据重编码装置，根据所述第一编码算法对所述至少两个经译码的位数据分组  $\{P_i\}$  中的每个分组进行重编码，产生局部码符

号数据分组 (Li)；和

比较装置，与所述数据重编码装置相连，把每个所述局部码符号数据分组 (Li) 与每个所述编码符号数据组 (Si) 比较，并对它们之间的差进行计数。

39. 如权利要求 38 所述的多速率译码器，其特征在于，所述译码器装置包含：

符号量度装置，根据第二编码算法把符号量度值赋予每个所述编码符号；

分支量度计算装置，与所述符号量度装置相连，根据相应的所述符号量度值产生一对分支量度值，它表示对应于来自所述输入缓冲器装置的所述每个编码符号的原始数据比特转换的加权概率；

判决路径装置，与所述分支量度计算装置相连，根据相应的所述分支量度值产生并存储对应于来自所述输入缓冲器装置的所述每个码符号的每次可能的原始数据比特转换的状态量度判决值；和

路径反向链接装置，与所述判决路径装置相连，对每个所述原始数据比特转换选择最可几的判决路径，并产生相应的所述经译码的数据比特。

40. 如权利要求 39 所述的多速率译码器，其特征在于，基本上以一块单片集成电路的形式来实现。

41. 如权利要求 40 所述的多速率译码器，其特征在于，所述码符号数据表示各所述经编码的原始数据比特的一串  $N_i$  个复制品。

42. 如权利要求 41 所述的多速率译码器，其特征在于，所述码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的  $(N_i - 1)$  个复制品序列。

43. 如权利要求 35 所述的多速率译码器，其特征在于，进一步包含：

在所述顺序译码器装置内的质量量度装置，产生每个所述经译

码的位数据分组 ( $P_i$ ) 的质量量度 ( $Q_i$ )，所述质量量度 ( $Q_i$ ) 符号数据差错与所述每个经译码的数据比特分组 ( $P_i$ ) 相关联。

44. 如权利要求 43 所述的多速率译码器，其特征在于，所述质量量度装置包含：

数据重编码装置，根据所述第一编码算法对所述至少两个经译码的位数据分组 { $P_i$ } 中的每个分组进行重编码，产生局部码符号数据分组 ( $L_i$ )；和

比较装置，与所述数据重编码装置相连，把每个所述局部码符号数据分组 ( $L_i$ ) 与每个所述编码符号数据组 ( $S_i$ ) 比较，并对它们之间的差进行计数。

45. 如权利要求 44 所述的多速率译码器，其特征在于，所述码符号数据表示每个所述编码原始数据比特的一串  $N_i$  个复制品。

46. 如权利要求 44 所述的多速率译码器，其特征在于，所述码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的 ( $N_i-1$ ) 个复制品的序列。

47. 如权利要求 35 所述的多速率译码器，其特征在于，所述码符号数据表示每个所述经编码的原始数据比特的一串  $N_i$  个复制品。

48. 如权利要求 35 所述的多速率译码器，其特征在于，所述码符号数据表示完整的经编码的第一原始位数据形式加上所述完整的经编码的第一原始位数据形式的 ( $N_i-1$ ) 个复制品的序列。

49. 如权利要求 35 所述的多速率译码器，其特征在于，所述译码器装置包含：

符号量度装置，根据第二编码算法把符号量度值赋予每个所述码符号；

分支量度计算装置，与所述符号量度装置相连，根据相应的所述符号量度值产生一对分支量度值，它表示对应于来自所述输入缓冲器装置的所述每个码符号的原始数据比特转换的加权概率；

判决路径装置，与所述分支量度计算装置相连，根据相应的所

述分支量度值产生并存储对应于来自所述输入缓冲器装置的所述每个码符号的每次可能的原始数据比特转换的状态量度判决值；  
和

路径反向链接装置，与所述判决路径装置相连，对每个所述原始数据比特转换选择最可几判决路径，并产生相应的所述经译码的数据比特。

50. 如权利要求 35 所述的多速率译码器，其特征在于，基本上以一块单片集成电路的形式来实现。

## 码分多址系统中提供译码位数据的方法及多速率译码器

本申请的主题涉及 Butler 等人于 1993 年 6 月 18 日申请、申请号为 08/079196、名称为“在通信接收机中确定传输的可变速率数据的数据率的方法和装置”的待批专利申请，该专利申请已转让给本申请的受让者。把该相关的申请通过援引全部包括在此。

### 技术领域

本发明一般涉及在有噪信道内对串行数字数据流进行译码的系统，尤其涉及以多种预定的数据率进行顺序译码的卷积维特比译码器。

### 背景技术

众所周知，在数字通信技术中，数据可以顺序地以连续方式或者以持续时间  $T$  恒定的帧形式进行发送，每帧的比特数  $N$  固定，它受到顺序传输速率与帧持续时间  $T$  乘积的限制。然而，在一些通信系统中，希望在所选的帧期间发送比  $N$  个数据更少的数据。这种系统之一是码分多址(CDMA)通信系统，它使用直接序频谱扩展技术。

在 Klein S. Gilhousen 等人的美国专利 5,103,459 中揭示了这种用于蜂窝式电话中的 CDMA 系统，该专利已转让给本申请的受让人。将专利 5,103,459 通过援引全部包括在此。在 CDMA 系统中，系统容量受到传输频带内的总干扰的限制。总干扰包括其它用户的信号，它主要由声码器数据(数字化的声音)组成。为了尽可能地减轻用户干扰程度，Gilhousen 等人所揭示的 CDMA 系统使用了可变速率的声码器数据，以适应在持续时间固定的数据帧内进行

在整个帧期间的数据传输。因此,对于全数据速率减小的帧,必需重复少于 N 个原始数据符号以用 N 个符号填满该帧。虽然,在相同的传输功率时对低速率数据的重复产生的干扰程度与全数据率的相同,但帧内的冗余信息可以使等价“信息功率”或每个符号的能量的传输功率相应地减少。这种技术降低了干扰程度,以使系统增加了多至如声码器冗余所允许的容量。

另一种做法,为了以减小的功率重复数据,可以仅在一帧内不减小传输功率地传输低速率数据一次。这种技术在该帧的一部分期间进行全功率传输,而在未使用的帧部分期间不进行传输。可以仅把帧内待传输的数据块放置在帧的开始部分,或者帧内其它预定的部分。然而,在具有很多用户的 CDMA 系统中,需要更复杂的块布局处理,以在整个帧期间对来自许多其它用户之间的干扰进行均匀地分布。在数据传输时的许多用户不必要的重叠将不必要地使系统干扰程度增加到最佳平均程度之上。

在 Gilhousen 等人于 1992 年 3 月 5 日提交、申请号为 07/846,312、名称为“数据子帧随机数发生器”的美国专利申请中揭示了复杂的块布局处理,以确保在整个帧内的干扰程度最小,该专利申请已转让给本申请的受让者,通过援引全部包括在此。

还可以参考美国加州圣迭戈的 Qualcomm 公司 1992 年 4 月 21 日出版的“推荐的 EIA/TIA 暂定标准:宽带扩展频谱数字蜂窝式系统双模式移动站—基站兼容标准”,以了解典型的 CDMA 系统。该 CDMA 系统标准规定了 5 种基本信道数据模式用于正向和反向链路通信。有 3 种正向链路模式和 2 种反向链路模式。根据特定的数据信道模式,规定了两种不同的卷积编码算法中的一种。两种是连续模式,三种是分组模式,顺序数据被组织到固定的帧时间间隔内。三种模式仅以几种预定的数据率中的一种数据率工作,而帧到帧的两种模式以四种不同的预定的数据速率中的任一种速率工作。

数据率和信道模式的这种多样性对任一种单译码装置提出了复杂的要求。在扩展频谱信道内进行实时串行译码使问题变得更困难。即,必须对接收到的信号进行实时检测、重新组合和译码。没有帧空间可用于传送原始声码器数据率的信息。很少有机会检查接收到的信号的几种可能的数据率,并决定应当对每一新的以哪种特定的速度译码。而且,在所有系统信道模式中还要用相同的译码装置,以避免部件增多。在上述的 Gilhouse 等人的专利中描述的 CDMA 系统使用了正交伪随机噪声(PN)编码、交织(interleaving)、带有对各二相移相键控(BPSK)符号进行正交覆盖的调制并对所覆盖的符号进行四相移相键控(QPSK)扩展以及对纠错的卷积译码。这么多种编码技术中的每一种技术都要求在每个信道模式的接收侧具有一定程度的译码能力。因此,对于为纠错而发送的卷积编码的符号进行串行译码的实时资源有强烈竞争。

对卷积码的译码技术是众所周知的,它包括由 A. J. Viterbi 提出的用于纠错的维特比算法, (“卷积码的误差限度和渐近最优译码算法”, IEEE 信息论汇刊, 1967 年 4 月, 第 IT-13 卷, No. 2, 第 260—269 页), 诸如 G. D. Forney, Jr. (“维特比算法”, IEEE 会刊, 1973 年, 第 16 卷, 第 268—278 页) 和 J. A. Heller 等人(“卫星和空间通信的维特比译码”, IEEE 通信技术汇刊, 1971 年 10 月, 第 IT-19 卷, No. 5, 第 835—848 页)等专家进行过讨论。

专家们已经根据各种具体的要求改进了维特比算法。例如, Hirosuke Yamamoto 等人(“带有重复请求的卷积码维特比译码算法”, IEEE 信息理论汇刊, 1980 年 9 月, 第 IT-26 卷, No. 5, 第 540—547 页)公开了带有重复请求质量量度的维特比译码算法。当接收机可在处理具有较长的约束长度而无反馈的卷积时与使用反馈信道来请求重传输时,他们的系统提供了同样的性能。Yamamoto 等人把一个质量比特加到译码器输出中,表示“帧质量”。如果是坏的,则请

求重复传输。他们证明,他们的算法的可靠性功能接近于没有重复请求的普通维特比算法的两倍。

同样,N. Seshadri 等人(“以卷积码作差错检测的扩广维特比算法”,全球通信 89 德克萨斯达拉斯,1989 年 11 月,第 1534—1538 页)提出了两种能应用于固定持续时间内的帧内编码的话音的推广维特比算法。在帧内容呈现出不可接受的差错程度的情况下,相邻话音帧之间的高度相关性可从相邻的帧估算出该帧的内容。Seshadri 等人证明,把奇偶比特加到话音数据中来进行高速率外循环块码的差错检测,可以获得非常可靠的帧内冗余。然后用内卷积码对增信帧进行编码,再对整个块进行调制,并经有噪声的信道发送。内推广维特比译码器为已译码的帧释放预定数目的候选对象,它们之中仅有一个应当有正确的奇偶信息。如果没有一个有正确的奇偶性,并且如果该信息位表示量化的话音帧,那么,能重新进行帧间估算,或者如果返回链路是可以用的,会进行自动的重复请求。只要正确的可选对象是预定数量的候选对象之一,则可以避免进行重传输。

遗憾的是上面的揭示既未讲授又未建议一种实时译码方法在不传输信息速率的情形下处理具有几种预定的数据传输率中的一种速率、并且速率可以每帧变化的一串帧。显然,在已有技术中存在未解决的问题和缺陷,本发明以下述方式解决了这一问题。

### 发明内容

本发明通过提供一种用单个超大规模集成(VLSI)器件实现的具有专用输入和输出缓冲器的串行维特比译码器(SVD)来解决上述问题。用维特比方法在帧模式和连续模式下对每一帧以所有预定的数据率对同步的和已量化的码符号流进行译码。设置了几种质量量度输出(“差错量度”)以确定数据率。本发明 SVD 的较佳实施例可以没有事先的数据率信息而对不同的数据率进行译码。

本发明的一个目的在于自动地以多种预定的数据率中的任何一

种速率对每一帧进行译码而无需发送速率信息。本发明的 SVD 对多种信道模式中的每一种模式实现了这一目的,通过设置多条通路以所有可能的预定数据率中的每一种速率进行译码,并通过提供“差错量度”来检测原来的数据率。本发明的 SVD 的优点是,只要卷积码已知和数据分组(或者帧,或者连续)模式已知,它就能以多种预定的帧数据率中的一种未知速率进行译码。本发明的 SVD 的另一个优点是,对于每个以假设的预定数据率被译码的每个帧,可以得到几个“差错量度”,用于建立原始数据传输率。

本发明的 SVD 通过记录译码后的输出数据流,把它与输入码符号流作比较,以估计输入流的串行差错率(SER),来提供这种差错量度。本发明的 SVD 可以包括诸如循环冗余检查(CRC)结果和 Yamamoto 质量量度(YQM)等其它的差错量度。SER 量度估计接收到的码符号内的差错。CRC 结果检测原始位数据内的位差错。YQM 表示在译码后的帧中估计的差错情况超出了预定的阈值。本发明的 SVD 的优点是接收机提供这些“差错量度”中的一个或更多个,作为不用从发射机接收速率信息而检测原始数据率的手段。

本发明的另一个目的在于对每种必要的信道模式,提供单个 SVD 装置进行译码。本发明的 SVD 可以把单个 VLSI 器件用于典型的 CDMA 系统的正向和反向链路信道模式中。例如,该 SVD 的一个较佳的实施例产生较高的译码增益,接近约束长度为 9 的速率  $1/2$  和  $1/3$  的卷码的理论极限,其中,在  $N=384$  个符号(速率  $\times 1/2$ )或  $N=576$  个符号( $1/3$ )的帧分组中处理码符号数据。可以用本发明的 SVD 对任何固定状态的帧分组的开始处和结束处或者对连续的码符号数据流进行译码。例如,在重复模式中,CDMA 信道这样工作,在必要时重复每个符号,以填满帧,把重复的符号聚集为本发明的 SVD 内的一个符号,以降低比特速率和功率。在反向 CDMA 链路数据子帧随机数发生器(DBR)模式中,仅把每一组重复的符号中

的一个符号用上面引用的 Gilhousen 等的专利申请中所描述的伪随机帧定比特技术来发送。用码速率选项和重复模式,本发明的 SVD 在正向重复模式或反向 DBR 模式中用任何预定的有效数据率对码符号流进行处理。

本发明的再一个目的在于把内部译码过程与外部信道定时要求隔开。本发明的 SVD 提供这样的手段来实现这一目的,向信道微处理器发送一个中断,并设置一个输出缓冲器,把微处理器与系统帧定时隔开,并保持译码数据和质量量度数据。本发明的 SVD 还包括输入缓冲器,要么允许码符号能连续地被 SVD 以信道符号速率接收,要么作为帧分组闯入 SVD。专用输入和输出缓冲器使本发明的 SVD 独立地操作外部信道和微处理器定时。灵活的微处理器接口使 SVD 可用于各种微处理系统。

如图 5 所示,本发明的 SVD 包括 5 个主要部件。输入缓冲器 (IB) 存储多于一帧的码符号数据。在块模式中,这可以使 SVD 必要时重复处理数据多次,以识别用于信道帧的实际的重复或 DBR 模式数据率。维特比译码器 (VD) 从 IB 接收软判决码符号,并以现有技术中已知的加一比较一选择 (ACS) 逻辑来对它们进行处理。把 ACS 处理的结果存储在 VD 内的内部路径存储器中。在通过许多判决字及时反向跟踪之后,通过该路径存储器的链式返回过程为每个码符号组提供了一个数据比特。把每个这些单个数据比特与诸如 Yamamoto 的质量量度 (YQM) 一起存储在输出缓冲器 (OB) 内,输出缓冲器 (OB) 是本发明的第三个部件。在帧分组模式中,以四种不同速率的对码符号进行译码,把得到的四种译码后的数据分组(包括质量信息)存储在 OB 内,并保持约半个固定的帧持续时间,以使微处理器能读取它们。第四个部件是控制块 (CB),它产生本发明的 SVD 工作所必需的所有内部定时信号。该内部定时从 CDMA 系统时钟和外部译码器同步选通信号得到。本发明的 SVD 通过第五个微处理器接

口(MI)部件来进行初始化和进行控制,它连接到本发明的第四个部件CB上。数据也可以通过MI来接收。

参考了下面的说明、权利要求和附图,将会使本发明的上述以及其它目的、特征和优点变得更明了。

#### 附图描述

为了更全面地理解本发明,现在参见下面结合附图所示的实施例的详细描述,其中:

图1是CDMA移动接收机的功能方框图;

图2是CDMA区站信道卡接收机的功能方框图;

图3包含图3A—3E,是五种CDMA信道类型的技术指标;

图4是普通的CDMA模式设置参数的技术指标;

图5是本发明的串行维特比译码器(SVD)的功能方框图;

图6包含图6A—6B,示出了维特比译码器的理论运作情况;

图7是本发明的输入缓冲器(IB)的功能方框图;

图8是CDMA系统的数据分帧随机数发生(DBR)符号选择的技术指标;

图9是本发明的维特比译码器(VD)的功能框图;

图10是本发明的输出缓冲器(OB)的功能方框图;

图11是本发明的Yamamoto质量量度(YQM)逻辑的功能方框图。

#### 本发明的实施方式

##### CDMA系统的译码问题

为了讨论和描述的目的,在上面引用的Gilhousen等人的专利中所描述的典型的CDMA移动通信系统的实施例的范围内描述本发明。然而,应当理解,本发明还可用于其它类型的通信系统,如,个人通信系统(PCS)、无线市话回路,用户小交换机(PBX),或者其它可用的无线通信系统。而且,使用诸如时分多址(TDMA)等公知的

其它传输调制技术的其它系统都可以使用本发明。如在上面引用的 Gilhousen 等人的专利中所描述的,典型的 CDMA 移动通信系统实施例对译码装置和方法提出了许多要求,直到现在,现有技术中已知的单个译码器设计仍未有效地满足这些要求。现在为解释的目的,对这些要求作如下简要的描述。

图 1 示出了 CDMA 系统的移动接收机调制解调器结构的方框图。本发明的串行维特比译码器(SVD)20 与移动站调制解调器 22 的其它部件有关。在工作时,RF 接收机接收射频信号,在模拟处理器 28 的参与和中央处理单元(CPU)30 的控制下,解调器 26 对射频信号进行解调。解调后的数据流由交织器 32 进行去交织,交织器 32 在 SVD 输入端 34 提供串行符号流。在合适的信号线上把同步、定时和时钟信号传送给 SVD20。SVD20 与微处理器总线 36 联接,用以与声码器 38 进行通信,声码器 38 对已由 SVD 译码的话音信号进行重建。

图 2 示出了涉及区站接收机调制解调器信道卡逻辑 40 的相同的 SVD20。CPU42、交织器 44 和解调器 46 在功能上与图 1 中与它们对应的部件相似。如图 1 所示,SVD20 在译码器输入端 34 上接收串行码符号流,并在微处理器总线 36 上产生并行数据流,用以把该数据流传送到逻辑 40 的其它部件(未图示)。

对于上面引用的 Gihousen 等人的专利中所揭示的 CDMA 系统,SVD20 必须以五种基本信道模式中的任一种进行工作,对正向和反向链路数据提供全部所需要的译码。在图 3A—3E 中描述了这些模式。从区站到移动用户的正向链路使用如图 3A—3C 所示的三种模式(同步、传呼和通话)。从移动用户到区站的反向链路使用如图 3D—3E 所示的两种模式(接入和通话)。

图 3A 通过例子描述了正向链路同步信道调制参数。从以速率 1/2(约束长度 K=9)卷积编码的信道对同步信道数据进行译码,并

对每个编码符号重复一次。数据以每秒 4800 个调制符号进行发送，SVD20 每 80 毫秒接收包含 384 个调制符号的超帧。因此，SVD20 能把该信道预先设定到固定的 1200bps 速率上。在帧边界上不用循环冗余检查(CRC)码把原始位数据编码成连续流。等效的原始位数据速率是 1200bps。

图 3B 通过例子描述了正向传呼信道调制参数。从以速率 1/2 ( $K=9$ ) 卷积编码的信道对传呼信道数据进行译码，并根据原始位数据速率，对每个码符号传输一次、两次或者四次。该信道的原始位数据率是预先设定的，并不随帧而变化。因此，SVD20 能把该信道预先设定到固定的原始位数据速率上。数据以每秒 19200 个符号进行发送，SVD20 每 20 毫秒接收包含 384 个调制符号的帧。在帧边界上不用 CRC 码把数据编码成连续流。

图 3C 通过例子描述了正向通话信道调制参数。从以速率 1/2 ( $K=9$ ) 卷积编码的信道对正向链路通话信道数据进行译码，并根据发端声码器对每帧选出的原始位数据速率，对每个码符号发送多至八次。在发端声码器的控制下，原始位数据速率能随帧改变，因为该数据速率不能预先设定，SVD20 必须以所有可能的速率对每帧进行译码。数据以 19200 个符号每秒(sps)进行发送，SVD20 每 20 毫秒接收 384 个调制符号的新帧。把数据编码成在开始和结束处为零状态(在进行编码之前，把“0”比特加在每帧的结束处)的分组，而 CRC 码预计出现在 96 个原始比特和 192 个原始比特的每个分组的结束处。并不希望 CRC 码带有短(48 和 24 比特)分组，因为空间的限制使 CRC 变得很花钱。

图 3D 通过例子描述了反向通话信道调制参数。从以速率 1/3 ( $K=9$ ) 卷积编码的信道对反向链路通话信道数据进行译码。对每个码符号重复多到七次(出现 8 次)，但在一个子帧内仅发送每个重复的码符号中的一个。子帧定时由取自前一帧 PN 码的最后几位的随

机数来决定。码符号重复率随每帧的原始声码器数据速率而变化。SVD 必须对每帧同时以所有的速率进行译码,这是因为发端声码器能随帧改变原始位数据速率。虽然 SVD20 并不知道一特定帧的原始位数据速率,但可从前一帧的 PN 码的最后几位得到重复符号的每一帧的子帧定时。数据以 28800 个码符号每秒进行发送,SVD20 每 20 毫秒接收包含 576 个可能的码符号(potential code symbol)的帧。把数据编码成在开始和结束处为零状态的分组,因为在每个分组的结束处加入了“0”比特。CRC 码预计出现在 96 个原始比特和 192 个原始比特的每个分组的结束处(紧接在“0”比特序列之前)。包含少于 96 个原始比特的帧省略了 CRC 码,以节省空间。

图 3E 通过例子提供了反向链路接入信道调制参数。从以速率  $1/3(K=9)$  卷积编码的信道对反向链路接入信道数据进行译码,并对每个码符号重复两次。原始位数据速率固定在 4800bps 上,而 SVD20 能为该信道进行预先设定,以工作在该单一的固定速率上。数据以 28800 个码符号每秒进行发送,SVD20 每 20 毫秒接收包含 576 个调制符号的帧。把数据编码成在开始和结束处为零状态(在每个分组的结束处加入“0”比特)的分组,但不提供 CRC 码。

本发明的 SVD 适用于图 3 所提供的五种示范性信道描述中的每一种,因为该 SVD 具有连续多速率译码、从连续到分组模式的转换和在( $1/2$  到  $1/3$ )卷积编码速率之间的转换的能力。更重要的是,本发明的 SVD 能对正向或反向通话信道数据中的每一种数据进行译码。参见上面引用的 Gilhousen 等人的专利和专利申请能更好地理解这种正向和反向通话信道的区别。

图 4 是一表格,该表格总结了本发明的 SVD 对于上面结合图 3 讨论的五种示范性信道所必需的控制信号功能。请注意,正向传呼和同步信道的原始位数据速率是固定的。

单片 SVD 的实施例

本发明的 SVD 最好是实现成单个超大规模集成(VLSI)电路。图 5 示出了 SVD20 的典型实施例的五种主要部件。输入缓冲器 (IB)48 存储 1.5 帧数据,因而,对一帧有多个译码通路,以在后面确定该帧的正确的重复或数据子帧随机数发生器(DBR)模式数据速率。维特比译码器(VD)50 在 7 比特符号缓冲总线(symbuf)7 上接收 IB48 的软判决符号。这些码符号由相加一选择一比较(ACS)逻辑进行处理,其结果作为状态量度存储在内部随机存取存储器(RAM)内。把 ASC 处理的判决存储到内部路径存储器内。在通过 64 级判决字及时反向跟踪之后,通过该路径存储器的反向链接处理识别各码符号组的单个输出数据位,以确保路径已与最可能的全球路径合并。这些输出数据位与质量量度(QM)信息一起通过数据线 56 存储在输出缓冲器(OB)54 内。在 VD50 译码完成之后,OB54 保留译码后的数据,以便微处理器接口 58 在译码数据(decdata)线 60 上进行访问。在分组模式中,以四种不同的原始位数据速率对码符号进行译码,并把得到的四个输出数据分组与相关的 QM 数据一起存储在 OB54 内。这种结构能使微处理器(未示出)用约 10 毫秒来读取 OB54 内的数据。通过微处理器接口 58 和产生 SVD20 所有必需的内部定时的控制器 62,对 SVD20 进行初始化和控制。定时从系统时钟和译码器同步选通得到。

#### 维特比译码算法

在现有技术中维特比译码器的一般理论是众所周知的,并通过参阅上面引用的一种参考资料可以理解。现在简要描述一下该理论以更容易理解本发明。

卷积译码器把原始数据比特序列(输入比特流)转换成编码符号序列(输出符号流)。对于每一个输入比特,有多个输出编码符号,它们由该输入比特和前( $K-1$ )个输入比特决定,其中  $K$  为译码器的约束长度。每个比特符号产生的编码符号数目由编码速率决定,即,例

如,速率为 1/2,数目为 2,速率为 1/3,数目为 3。根据诸如由序列  $X^8 + X^6 + X^5 + X^4 + 1$ (八进制值 0561<sub>8</sub>)表示的速率 1/2G1 码的专门的多项式码,通过对输入流进行移比特和异或产生每个码符号。多项式码内的比特数与约束长度相等,对于本发明的 SVD 较佳实施例,该值固定为 9。通过模拟非系统码来选择实际码(G0,G1)或(G0,G1,G2),以确定移动环境内最佳的差错特性。上面讨论的在典型的 CDMA 系统中使用的较好的卷积码是:对于正向链路,G0=0753<sub>8</sub> 和 G1=0561<sub>8</sub>;对于反向链路,G0=0557<sub>8</sub>,G1=0663<sub>8</sub>,G2=0711<sub>8</sub>。这些卷积码提供为 1/2 编码速率提供了最小的自由汉明(Hamming)距离 12,为 1/3 编码速率提供了最小的自由汉明距离 18。

维特比译码器算法通过为输入码符号流寻找最可能的译码序列进行运算。首先,计算每条可能路径的相对概率的状态量度或权重。把最可能进入每个状态的转换存储在所有状态的路径存储器内,然后译码器通过最可能的序列及时跟踪或反向链接,以选择每个输出位。该过程中主要的步骤是分支量度产生、状态量度产生和反向链接路径判决。三个参数决定了特定的约束长度和速率的译码器的性能:输入码符号的步长和量化级数目、状态量度归一化方法和路径存储器的有效反向链接深度。

分支量度是对应于转换概率算法的费用函数,每个可能的比特转换已经产生了给出的软判决输入码符号。对软判决内的符号和幅度信息,进行度量和组合以产生每个分支量度。对于 1/2 速率,有四种可能的量度。对于 1/3 速率,有八种可能的量度,对于任意的 1/n 速率,有  $2^n$  种可能的量度。通过经验和模拟,在目标信道上为最好的译码器性能选择用于计算和度量分支量度的公式。这些公式由本发明的 SVD 逻辑来实现。量度输入由存储在 VD50 的符号量度表(SMT)内的符号量度值来换算。“删除”电平使在计算分支量度时忽略码符号。删除可用于在需要较高的信道数据速率时“压缩”码。另

外,删除还能用于除去一些属于“隐藏”数据或控制信道的信道码符号。由于被删除的码符号不再影响量度值,所以适当的删除率对译码性能并无显著的影响。

卷积码能产生无穷的码符号序列,但码的某些特性使它可以减少符号序列的数目。第一个特性是仅考虑进入状态的最佳(最有可能)的路径,因为通过一状态的任何全局路径必须采用最佳的局部路径。第二个特性是卷积码结构是重复的,它有对称的码树。因此,必须把码序列并入由数目有限的独有的比特流图形产生的等效序列中。对于给定的约束长度  $K$ ,有  $2^K - 1$  种可能的数据比特图形(此处命名为“状态”),必须对它进行估计,以确定最有可能的全局路径。

图 6 示出了基于  $G_0=5_8$  和  $G_1=7_8$  的例示的  $K=3(1/2)$  码合并序列的典型篱笆图。图中示出了在状态之间进行转换所需要的符号。对于每种状态,要计算表示沿通过该状态的一条路径的相对概率的状态量度(未示出)。状态量度计算由相加一比较一选择(ACS)过程来进行。把当前状态之前的每种可能的状态的状态量度加到分支量度中,以前一状态转换到当前状态。把和进行比较,选出由最小的和所表示的最可能的转换,并把它分配给当前状态作为状态量度。每个 ACS 的判决比特是产生所选转换的前一状态的最低有效比特(最老的比特)的值。对篱笆列上的所有状态的判决成为路径存储字。

由于发生器多项式( $G_0$  和  $G_1$ )内的第一项和最后一项是相同的,在两条路径上发送的进入和离开任一种状态的符号( $C_0, C_1$ )的假设(hypothesis)( $i, j$ )是二进制的。这些 ACS 状态关系用图 6B 的蝶形图来图示。当然,在无噪声信道中,码符号值( $c_0, c_1$ )是无差错的,状态量度为零,或者在最大值处饱和,零状态量度表示最可能的全局路径上的一个状态。

参见图 6B,把从状态  $x_0$  到  $0x$  的分支量度加入到  $x_0$  状态量度上,以确定两种可能的  $0x$  状态量度的第一种。把从状态  $x_1$  的分支量

度加到  $x_1$  状态量度值上,以寻找第二种可能的  $0x$  状态量度。然后把这两种可能的值中最小的值作为新的  $0x$  状态量度值。对  $1x$  状态和所有该篱笆列中的其它状态重复这种过程。对应于一个原始位为每个新码符号组产生新的列。把每对状态量度之间的差值与预定的质量阈值(QT)作比较,以 Yamamoto 等人在上面引用的论文中所建议的和如下文结合图 12 所描述的方式得出 Yamamoto 质量量度(YQM)或新状态的“qbit”。

一当把局部 ACS 判决的矩阵存储到路径存储器内时,反向链接过程沿反向通过该矩阵的路径。反向链接在由 ACS 阵列报告的“最佳状态”处开始,然后用在路径存储器字内的该状态(比特地置)的判决来确定前一最佳(最可能)状态。通过至少五或六个判决的约束长度来运行反向链接,以保证采用的路径并入最可能的全局路径。对于约束长度为 9,有 63 种状态的反向链接路径深度是足够的。反向链接结束处的最后的判决被认为是维特比译码器的输出位的最佳判决。对于每个顺序的原始数据位,由 ACS 阵列产生新的判决字和新的最佳状态,并通过相同的路径长度从新的篱笆列返回进行重复反向链接过程。因此,每个新的比特符号通过一个篱笆走出一个路径存储器内的 64 状态反向链接窗。

在上面结合图 3 讨论的示例性的帧规格中,对于把数据分成组的所有操作模式,通过把八个“0”比特的尾部插在数据的结束处,而在所有“零”状态时启动和结束译码。在多速率译码时,SVD 在每帧的第一组八个篱笆列为零期间把判决比特强制输入到路径存储器中。这保证了为每帧所选的路径在所有可能数据速率的所有“零”状态时启动和结束。

#### 输入缓冲器(IB)的描述

下面对本发明的 SVD 功能的描述依赖于对各种处理过程的这些示范性的定义。

内部 CHIPX8 时钟:这是内部器件工作的主时钟,最好为 9.8304MHz。

计算周期:这是通过 ACS 对 108(图 9)处理分支量度和各对先前状态量度的时间,等价于两个内部 CHIPX8 时钟周期。

处理周期:这是通过反向链接处理一个原始数据比特的时间,等于  $128+3$  个计算周期,对它的要求是能通过 ACS 逻辑 106 处理完 64 个分支量度。

块周期:这是对四种可能的帧分组大小的处理时间,等于 192、96、48 或 24 个处理周期。

缓冲器周期:这是在 VD50 中处理一帧所有四种可能的分组速率并形成最终的反向链接清洗和清除操作的时间。它等于 432 个处理周期( $192+96+48+24+72$ )。

帧:这是发送分组内的所有码符号所需要的时间窗,除了同步信道之外,一般等于 20 毫秒,累积三个 26.67 毫秒的帧形成一个 80 毫秒的超帧。

图 7 提供了本发明的输入缓冲器(IB)48 的较佳实施例的示意框图。IB48 在译码器输入线 34 上接收码符号,并在 symbuf 总线 52 上提供为所有可能预定的声码器数据速率已进行选择和累积的这些码符号。如图 4 所示,IB48 对正向链路信道工作在重复模式中,而对反向链路信道工作在重复模式或数据子帧随机(DBR)模式中。在重复模式中,IB48 把码符号进行累加,使它们以采用最初的全速率的输入码符号速率的  $1/2$ 、 $1/4$  和  $1/8$  的累加符号速率提供在 symbuf 总线 52 上。在 DBR 模式中,IB48 根据从按照最初全帧块分组的当前帧内的 PN 序列的最后几位得出的随机产生的码字,为按位于帧内  $1/2$ 、 $1/4$  和  $1/8$  的数据块大小分组选择码符号。两种 IB 模式在译码器输入线 34 上输入的码符号是相同的。当 IB48 已接收到足够多的码符号以跟上 VD50 译码处理时,由 DECSTB 线 64 选通编码符

号位,在 INBUFRDY 线 66 上向 VD50 发送准备选通信号。然后 IB48 根据要求经 symbuf 总线 52 向 VD50 顺序提供码符号。

由 DECMODE 总线 68 上的模式控制字控制译码器模式,它包括几个影响 IB 工作的控制比特。各种 IB48 的工作模式可以参照图 4 来理解。在 DECMODE 总线 68 上的信号中的几个信号还决定在 inbufrdy 线 66 上向 VD50 发送的选通信号的定时关系。这些位中的一位对应于如上面结合图 3 所描述的四种可替换的原始位数据速率的四种不同的分组大小中的每一种确定码符号边界。DECMODE 总线 68 上的其它一些位根据 DBR 位置码决定是否在将重复的码符号提供至 symbuf 总线 52 或仅选出它们之前由 IB48 对它们进行累加。

以 DECSYNC 线 70 上出现的选通信号开始每个 SVD20 的输入周期。在 DECSTB 线 64 上的允许对 IB48 进行初始化的分组的第一个码符号的选通信号出现之前的至少 15 个内部时钟(CHIPX8)在 DECSYNC 线 70 上出现选通信号。在线 70 上的每个选通信号之后,IB48 要有 384 或 576 个码符号通过 DECSTB 线 64 被选通进入缓冲器 72。卷积编码速率(图 4)决定是  $384 (=2 \times 192)$  个还是  $576 (=3 \times 192)$  个码符号。由内部 CHIPX8 时钟(未图示)把译码器输入端 34 上的码符号位从最高有效位(MSB)到最低有效位(LSB)连续地加以串行计时。由 DECSTB 线 64 上的选通信号对每个符号的最低有效比特作标记,从而把所有码符号锁存入并行符号寄存器 74 内。如果码符号来自卷积去交织器,则由 DECSTB 线 64 以每帧 384 个符号的速率对输入符号计时。如果工作在反向链路信道,码符号来自块去交织器,以高至最大输入速率(每七个内部 XHIPX8 个时钟周期一个码符号)使输入码符号在 IB48 内分组。最大分组速率受到仲裁逻辑(未图示)的限制,它允许在填充时从缓冲器 72 读取码符号。当在 symbuf 总线 52 上开始输出时,缓冲器控制逻辑 76 在 INBUFRDY 线 66 上产生信号,以通知控制器 62。

缓冲器控制器 76 内的两比特“psize”计数器(未图示)由线 66 上的 INBUFRDY 选通信号进行复位。psize 计数器的内容表示 symbuf 总线 52 上的码符号的分组大小,它可以是 24、48、96 或 192 个原始比特。线 66 上的选通信号启动 VD50 开始工作,用 DECMODE 总线 68 上的两个比特来指示在线 66 有效之前 IB48 必须接收到的码符号的数目。这种可选择的 INBUFDY 线 66 的选通延时使 SVD20 内处理延时优化。

当选通 INBUFDY 线 66 后,就使缓冲器 72 的读地址和重复缓冲器 78 的写地址复位。然后预取缓冲器 72 内的第一个码符号,并把它锁存在 symbuf 总线 52 上。然后响应于 symstb 线 80 上的一个选通信号,使缓冲器 72 的读地址指针加 1,取出下一个码符号,并把它锁存到总线 52 上。每七个内部 CHIPX8 时钟周期 symstb 线 80 上的 IB 读选通被限制到一个读周期,以有时间对缓冲器 72 的读和写操作进行交织。从当前帧读取的码符号的数目由 DECMODE 总线 68 上的信号确定。在缓冲器 72 从第一分组读取了表示 192 个原始数据比特的码符号后,根据选择和累积逻辑 82 从重复缓冲器 78 读取 96、48 和 24 个位分组。在读取了每个分组的最后一个码符号后,使重复缓冲器 78 的读和写地址复位,缓冲器 76 内的“psize”计数器(未图示)加 1。“psize”计数器的内容决定了从缓冲器 72 或重复缓冲器 78 取出的码字地址,而且还控制选择和累积逻辑 82,使合适的码符号出现在 symbuf 器总线 52 上。在每次读取之后,只要“psize”计数器不为零,就将重复缓冲器 78 的读地址指针加 1。在从缓冲器 72 或重复缓冲器 78 读取了每个其它数据之后,把重复缓冲器 78 的写指针加 1。以相同的顺序从 IB50 输出码符号而不考虑 SVD20 的工作模式,但当在连续模式中以固定的重复速率工作时,VD50 忽略未使用的分组。

当在重复模式中从缓冲器 72 读取码符号时,由逻辑 82 把每一

对码符号加在一起，并存储到重复缓冲器 78 内。当 INBUFDTRY 线 66 被选通并且在把每一对之和写入到重复缓冲器 78 之后，将累加器复位到零。当从重复缓冲器 78 读取码符号时，它们也以对的形式相加，并写回到重复缓冲器 78 内。对 192、96 和 48 个符号分组的每个符号对重复这种相同的操作。重复缓冲器 78 的 7 个比特的字长度可以存储而不用对多到 8 个编码符号的和进行截断。当工作在 DBR 模式时，强使反馈到逻辑 82 内的经累加器累加的码符号为零，从而逻辑 82 的输出是从缓冲器 72 的输入的符号。仅允许由 DBR 码比特（未图示）所选的符号写入到重复缓冲器 78。psize 寄存器的内容和行地址（发送的时隙数）控制把哪个 DBR 码比特用于选择符号。如此选择一组 DBR 码比特，使任何低速率的 DBR 码比特成为高速率 DBR 码位的子组。

正向链路同步信道使用 128 个符号的 26.67 毫秒的帧，这些符号以不变的每秒 4800 个符号作为连续流发送。IB48 在 80 毫秒内接收三去交织同步帧，使一个译码器分组具有 384 个符号，但为这种“超帧”仅需要在 DECSYNC 线 70 上从去交织器得到一个选通信号。对每个同步信道码符号重复两次，并把该 1200BPS 信道上的 80 毫秒的帧译码成带有符号重复因数为 2 的 96 比特分组。同步信道工作在连续模式，但由于通过 SVD20 延时的缘故输出数据延迟 71 个比特，每个分组中的第一位先于当前分组接收到的 6 个分组的 26.67 毫秒的同步分组的第二十六位。这些定时细节由控制器 62 和微处理器接口 58（图 5）来控制。

在反向链路信道，能以高至片码速率把 576 个符号帧一次送入 IB48 内，该片码速率是 CHIPX8 时钟速率的八分之一。把码符号按顺序写入到缓冲器 72 内，它可以被看作 32 行×18 列的阵列。按列读取和写入码符号。即，以与写入它们相同的顺序从缓冲器 72 读取码符号。当接收到一个分组中的第六个符号时，IB48 可以在每个处

理周期期间向 VD50 发送三个一组的码符号(表示一个原始数据比特),直至处理完整个分组。在复位与每个缓冲周期结束之后,由内部时钟把译码器输入端 34 上的由 DBR 选择的比特连续计时到 DBR 编码寄存器(未示出)内。在 DECSYNC 线 70 上有选通信号时把结束处带有 DBR-13 的十四个比特作为下一个缓冲周期的 DBRCODE 字锁存。这些 DBRCODE 位,DBR-0 到 DBR-13 用于选择经总线 52 送至 VD50 的全速率符号写入到重复缓冲器 78 内的码符号。一当处理完整个分组,就同样的方法处理留在重复缓冲器 78 内的半个分组。在经总线 52 向 VD50 发送半个分组的码符号时,把那些由 DBRCODE 选出的符号作为四分之一个分组重新写入到重复缓冲器内。用类似的方法处理四分之一分组和下一个八分之一分组。

图 8 示出了用于确定从缓冲器 72 用哪 16 个时隙(行对)的 DBRCODE 符号选择算法。对于全速率操作,发送所有 16 个时隙(32 行)。然而,在较低的速率下,发射机在传输一些时隙期间关闭。例如,四分之一速率仅发送第一组四个时隙中的一个,作为第一组 36 个码符号。按行来发送码符号,但去交织器按列把它们发送给 SVD20。由于发送 DBR 算法在交织器内跳过一些行,所以部分速率实际上没有几行。对于 DBR1/4 速率,缓冲器 72 仅包含八行数据和余下行中的噪声符号。与反向链路通话相比,反向链路访问信道(图 3E)以 1/3 卷积编码速率工作在重复模式,码符号重复两次,以在反向通话信道上提供约 2.5dB 的增益。参见上面引用的 Gilhousen 等人的专利申请能更好地理解发送 DBR 算法。

除了图 7 所示的功能逻辑之外,IB48 还包含测试逻辑,以提供自测试验证功能。

#### 本发明的维特比译码器元件

图 9 为 VD50 的功能框图,VD50 处理从 IB48 到达总线 52 上的码符号。VD50 在 rdata 线 56 上与质量信息一起输出译码后的比特

流。这些功能在图 9 所示的六个子模块中完成。DECMODE 总线 68 来的两个比特控制 VD50 在 DECRATE 线 84 和 PACKET 线 86 上的工作。存储在符号量度表(SMT)88 内的数据表为可编程的查找表,它把总线 52 上的 7 比特码符号转换成合适的 4 比特标度,以便分支量度逻辑 90 进行校正操作。SMT88 提供了处理不同操作模式所需要的灵活性。对 VD50 的控制信息一般设在分组处理呼叫的开始处,在通话接收期间不变化。线 84 上的 DECRATE 信号决定在分支量度计算中包括多少个码符号,并且,线 86 上的 PACKET 信号在各分组开始时清除状态量度值。SMT88 的内容把总线 22 上的 4, 5, 6 和 7 比特输入码符号转换成用于分支量度逻辑 90 在内部 SMT-SYM 总线 92 上的经定标的四比特输出码符号。这些转换包括对累积在 IB48 内的低数据速率分组的码符号进行加倍、四倍和八倍操作的必要的补偿。在总线 52 上输入的每个码符号为 SMT88 提供了低阶地址位,然后把诸如地址等数据作为 SMTSYM 值在总线 92 上输出。SMT88 为重复和 DBR 两种模式中四种速率的每一种存储单独的基准点。

对于每种分组,VD50 保持对表示 Yamamoto 质量量度(YQM)的一个质量比特的跟踪。把每个各分组大小的最近处理周期的最佳状态(定义为零)的质量比特存储在 YQM 寄存器 93 内,该质量位在后面用于确定原始数据传输速率。

符号差错率(SER)逻辑 94 把输入 i 和 j 符号硬判决与由重译码输出数据的 c0 和 c1 值作比较,以在 SERROR 总线 96 上产行每个分组的 SER 字节。总线 96 上的值最大(饱和)为 255,该值也在后面用于确定原始数据传输速率。

SMT88 由随机存取存储器(RAM)和负载控制逻辑组成,向 RAM 多路传输写地址和写选通信号。SYMBUF 总线 52 上的 7 比特输入符号用于在 SMTSYM 总线 92 上成为输出符号的 4 位数据

值的地址的 LSB。psize 总线 98 上的 psize 值形成 RAM 地址的两个 MSB，它能为不同的分组选择不同的转换来补偿由 IB48 引入的累加和。

用速率 1/2 码的一对(c0,c1)和速率 1/3 卷积码的三个一组的(c0,c1,c2)对维特比译码器据以工作的篱笆的分支(图 6A)作标记。因此，在任何 ACS 操作之前，必须分别计算速率 1/2 和 1/3 的两种可能的值(c0,c1)或三种可能的值(c0,c1,c2)的每个值的适当的分支量度。这些分支量度根据下面的公式 1 来计算。例如，在每个计算周期期间，分支量度逻辑 90 在 BMETRIC 总线 100 上产生一对新的量度 R<sub>ijk</sub>。它们是根据 SMTSYM 总线 92 上的输入符号(r<sub>2</sub>, r<sub>1</sub>, r<sub>0</sub>)和定时控制逻辑 104 在 hyp 总线 102 上发送的符号(c<sub>2</sub>, c<sub>1</sub>, c<sub>0</sub>)的假设(hypothesis)计算的。

$$R_{ijk} = \sum_{ijk} [(\overline{r_{x3} \oplus c_x}) * (4 * r_{x2} + 2 * r_{x1} + r_{x0})] \quad [\text{公式 1}]$$

其中 c<sub>x</sub> 为目标 ACS 状态时总线 102 上的 i,j 或 k 的假设，r<sub>x3</sub> 为总线 92 上的每个码符号的记号，(r<sub>x2</sub>, r<sub>x1</sub>, r<sub>x0</sub>) 为 SMTSYM 总线 92 上的每个符号量度的三个 LSB。

把总线 92 上表示单个原始数据比特的三个符号量度中的每一个选通入在分支量度逻辑 90 内的一组输入锁存器(未图示)内。为译码速率 1/2(或 1/3)转换输入信息的两个(或三个)码符号，在总线 100 上提供四比特分支量度对。如果 DECRATE 线 84 指定卷积编码速率为 1/2，则强制使总线 92 上的第三符号为零。把总线 100 上的分支量度对送到相加一比较一选择(ACS)逻辑 106，其中，在每个 ACS 计算周期期间选出两个分支量度中的一个。对于关于输入数据的每个假设，根据公式 1 计算总线 100 上的 4 比特分支量度。还产生

一个4比特分支量度用于实现假设。在该式中,如果码符号的记号与该假设一致,则把码符号的幅度加到  $R_{ijk}$  量度上;否则,把零加到量度上。速率  $1/2$  的分支量度值预计在 0 到 14 的范围,而速率  $1/3$  的分支量度预计在 0 到 15 的范围。ACS 逻辑分支量度加法器的饱和值为 15。

从 SMT88 到分支量度逻辑 90 的总线 92 上的符号量度以记号一幅度的格式表示,由符号值根据下面示范性的查找表来确定删除。

速率 $1/3$ 的符号量度	速率 $1/2$ 的符号量度
最强的一 0101	最强的一 0111
最弱的一 0001	最弱的一 0001
删除 0000	删除 0000
删除 1000	删除 1000
最弱的零 1001	最弱的零 1001
最强的零 1101	最强的零 1111

如果速率  $1/3$  的预计范围外的符号量度溢出了该累加器,则总线 100 上的分支量度在  $15(1111_2)$  饱和。

ACS 逻辑 106 在每次允许的计算周期期间,处理总线 100 上的输入分支量度。一对 ACS 逻辑块 108 处理总线 100 上的 4 比特分支量度和状态 RAM110 来的 5 比特状态量度,以在内部总线 112 上产生一对判决位,并在另一根内部总线 114 上产生一对新的状态量度。把在总线 112 上的判定位(它们是每个 ACS 对 108 中的最佳先前状态的 LSB)移入 8 比特输出锁存器 116。在判决(desisions)总线 118 上把每四对判决对的 8 比特判决字节输出到反向链接逻辑 120。

在每个分组的第一处理周期期间,在从状态 RAM110 读取时,把零状态的量度设置成零,而把所有其它量度设置到它们的饱和值。在分组模式中,对于每帧的开头 8 个处理周期,强使总线 118 上的判决和总线 122 上的最佳状态为零。在通过把当前状态从 MSB 移到

LSB 计算得到的位置上把新的状态量度、前一状态量度的最小和以及链路分支量度写回到状态 RAM110 中。为相等的量度选择零假设判决。在每个处理周期开始时把状态零的新的量度写入到最佳状态锁存器 124 中。把在当前处理周期中的所有其它新的状态量度与锁存器 124 内的当前最佳状态量度相比较，并用较小的一个量度代替锁存器 124 内的当前最佳的状态量度。在当前处理周期期间当从状态 RAM110 读取前一处理周期原最佳状态量度时，从每个状态量度中减去前一处理周期的最佳状态量度。ACS106 继续累加量度并进行归一化，直到所有四个码符号分组都被处理。

把 Yamamoto 质量量度(YQM)或“qbit”依附于各状态量度。在第一个 IB 处理周期期间，把初始状态零的 YQM 比特设置成“好”或真(0)，把所有其它的 YQM 比特设置成“坏”或假(1)。如果选出的先前状态量度的 YQM 位是假，或者 ACS 计算得到的量度差小于或等于量度阈值寄存器(见图 11 中的 MUX216)规定的预定的质量阈值(QT)，则把 ACS 逻辑 106 计算得到的每个新的状态量度的 YQM 比特设置成假(1)。

本发明重要组成部分是在译码期间产生的“差错量度”组，它包括三类质量信息。对于几个数据速率假设(例如，9600, 4800 等)，SVD20 产生一个或一个以上个独立的质量量度元素。这些包括循环冗余检查(CRC)结果、符号差错率(SER)和 Yamamoto 质量量度(YQM)。此处把 YQM 还取命为“qbit”。在低数据速率时省略了一些差错量度。在现有技术中 CRC 和 SER 是众所周知的。参照上面引用的 Yamamoto 等人的论文和下面结合图 11 所作的讨论可以理解 YQM 量度。差错量度用于以上面引用的 Buler 等人的专利申请中所揭示的方法选择最可几原始传输数据速率。

为每个数据速率假设产生译码符号数据，并把该数据存储在 OB54 内。每个假设的帧数据包括三个质量量度元素。下面结合图 10

讨论此处的CRC元素。下面结合图9内的SER140讨论此处的SER元素。图11示出了产生YQM比特的ACS对108(图9)的方框图。

回到图6B,把从状态x0到状态0x的分支量度加入到x0状态量度中,以确定两个可能的0x状态量度中的第一个。通过把状态x1分支量度加到x1状态量度值中来寻找第二个可能的0x状态量度。然后把这两个可能的值中最小的一个值指定为新的0x状态量度值。对于篱笆列中的1x状态所有其它的状态重复这一处理过程。YQM位是一标记,它显示出两0x状态量度值之间的差值小于预定的质量阈值(QT),或者,它采用另一种小于预定的QT的状态。在图11中,在线210上设置ACS0x0状态的LSB,在线212上设置ACS0x1状态的YQM比特。在多条4比特总线上为多路复用器MUX216设置QT阈值组214。psize总线218根据当前传输数据速率假设选择QT组214中的一个。本发明的一个重要原理是组214中的每个质量阈值(QT)是可编程的,并能独立地或一起进行调整,以适应不同的工作模式。由于不同的速率假设可以有不同的QT,所以MUX216在组214的可选用的QT项之间切换。

继续参见图11,计算两个ACS状态量度之间的差值,并在总线220上向四路XOR(异或)门222提供该差值。线224上的判决比特用于在多路复用器MUX226中选择两个量度中最可几量度和在总线220上启动MET0DIF。线228上MUX226的输出表示由总线210和212所选的状态量度比特。把输出线228与线232上的比较器230的输出相“或”,以在线234上产生YQM比特。比较器230把从组214选出的QT与总线220上的状态量度差作比较(在调整了XOR门222的正负之后)。把YQM比特234送至YQM寄存器93(图9)。

维特比篱笆(图6A)内的当前状态的YQM比特以这种方式产生。YQM过程具有强制使维特比篱笆的任一状态的“坏”YQM比特

通过判决树向前传播。YQM 比特 234 是一个指示两输入的状态量度之间的差值大于或小于所选的 QT 值的标记。如果该差值大于 QT 值，则把 YQM 比特设置成前一状态的 YQM 比特。如果该差值小于所选的 QT 值，那么把新状态的 YQM 比特设置成“坏”或“1”。在任何速率假设的译码操作开始时，用“好”YQM 比特（“0”）对零维特比篱笆状态作标记，用“坏”YQM 比特对所有其它的篱笆状态作标记。在速率假设的译码过程结束时，根据存储在寄存器 93 内，并从 ACS 逻辑 106（图 9）输出的最后的零状态的 YQM 比特用“好”或“坏”对帧作标记。因为每个声码器帧完成时都带有八个零（00000000），所以译码器知道帧的正确的最后状态为零状态。

反向链接逻辑 120 包含 64 个字的路径存储器 126，每个字有 256 个判决比特。在每四个计算周期期间，读取路径存储器 126 二次，然后写一次。因此，在每个处理周期期间，反向链接逻辑 120 能把 256 比特写入到一个路径存储器字内，并从 64 个这种路径存储器字的每个字读取一个比特。反向链接逻辑 120 包括多路复用器 128、地址产生器 130 和多个数据锁存器 132。地址产生器 130 确定写入字的地址，一个处理周期的第一次反向链读取在前一写字操作时开始。

反向链接处理是由按自确定序列写和读路径存储器 126 来完成的。对总线 118 上的最近的判决字，把读地址的八个 LSB 与总线 122 上的最佳状态一起启动。对于最后的字（因为最佳状态的输入强制为零）和每个块周期的开头八个字（因为判决和最佳状态输入在那里强制为零），该比特地址为零。把对每个被读取字存储在位地址处的判决移到作为下一个读字操作中的位地址的 LSB 的读地址内。把形成进入判决字的比特地址的读地址的八个 LSB 循环一位，把 MSB（第八位地址）移入 LSB 的位置。这一循环补偿了由 ACS 对 108 作的判决排序。在每次从路径存储器 26 内的最新的到最老的判决字读至反向链接部件之后，地址产生器 130 减小字地址。第 63 次读取操作时

的判决比特是内部比特线 134 上输出到输出数据锁存器 132 的位。通过 432 个处理周期来处理每个帧,包括 192 个全数据速率周期,96 个半数据速率周期,48 个四分之一数据速率周期和 24 个八分之一数据速率周期。

在分组和连续两种模式中,除了上述的最佳状态控制之外,反向链接操作完成相同的功能。线 134 上的输出比特由 rdata 线 56 上的反向链接逻辑 120 来计时…[如前]。

符号差错率(SER)逻辑 94 对 rdata 线 56 上的输出数据流进行再编码,并把重新产生的码符号与 SVD20 在译码输入线 34 上接收到的码符号比较。SYMSTB 线 80 上的选通信号把总线 92 上的每个符号量度的符号比特 r3 计时入先进先出存储器(FIFO)136 内,它补偿 VD50 内的线路和反向链接延时。在每帧开始准备把线 56 上的每位的计时入编码器 138 内的 9 比特移比特寄器时将编码器 138 和差错计数器 140 复位。在 8 比特差错计数器 140 内对未比较的码符号的数目进行计数,该计数器的饱和值为 255。把计数器 140 的输出在 SERROR 总线 96 上提供给 OB54,在那里它被移位和锁存入合适的寄存器内。

当定时控制逻辑 104 在 INBUFERDY 线 66 上接收到信号时,逻辑 104 使 VD50 开始工作,然后使 VD50 其它部件之间的定时相协调。逻辑 104 在 SYMSTB 线 80 上产生两个选通信号(对于速率 1/3,为三个选通信号),以把数据通过 SMT88 计时入分支量度逻辑 90 和 SER 逻辑 94 内。在每个计算周期期间,VD50 与 ACS 对 108 一起处理 hyp 总线 102 上对一个假设的当前原始的数据位符号。处理周期包括处理每种 ACS 状态的每个位符号的 128 个计算周期。因为本实施例的卷积编码器的约束长度为 9,每个原始的数据位符号(在每个篱笆列内)有  $2^{9-1} = 256$  种状态。定时控制逻辑 104 内的状态序列计数器对 256 状态进行计数(每个计算周期 2 种),使计数值

出现在 STATESEQ 总线 142 上。定时控制逻辑 104 内的其它部件对状态寄存器的内容进行编码,在 hyp 总线 102 上产生假设。其它的控制信号提供在 ACSSTATE 总线 144 上,以(a)使 ACS 处理空转和使逻辑 106 空闲,(b)在第一处理周期期间使输入到 ACS 对 108 的状态量度为零,(c)在每个处理周期的结尾处作标志以能储存最佳状态,(d)为每个计算周期启动判决锁存器 116。

定时控制逻辑 104 确定路径存储器 126 的读和写地址,并把它们分别提供在地址总线 146 和 148 上。每次写操作时,逻辑 104 内的 11 比特计数器(未示出)加 1,以在总线 118 上产生每个判决字节的写地址。在一处理周期期间,每四个计算周期发生在总线 118 上进行写判决字节的 32 个操作中的一个操作。由产生读选通信号和在写选通之间两次减小地址的同一个逻辑在总线 146 上产生读地址的 6 个 MSB。根据逻辑 104 内的定时关系,产生使读地址的 LSB 在总线 146 上装载、移位和变零所需要的控制信号。该定时关系还确定 RSTB 线 150 上的 RSTB 选通信号,以使反向链接逻辑 120 和 SER 逻辑 94 的工作相协调。定时和控制逻辑 104 还产生将 SER94 内的计数器 140 复位和启动的定时信号。

#### 本发明的输出缓冲器(OB)

图 10 示出了输出缓冲器(OB)54 的功能方框图。在转换器逻辑 152 内把 rdata 总线 56 上译码后的数据转换成字节,并把它们存储在缓冲器 154 内。转换器 152 还以通用的方式检查分组 CRC 码。在通过 INTD 线 156 向微处理器接口 58 开始发出中断约 10 毫秒后能在 DECDATA 总线 60 上得到至微处理器接口 58 的分组和分组的状态字节。把 CRC 对分组的检查结果存储在状态寄存器(未图示)内。转换器 152 所用的 CRC 多项式取决于分组的大小,具体如下:

分组大小	CRC g(x)
24 比特	无

48 比特	无
96 比特	CRC $x^8 + x^7 + x^4 + x^3 + x + 1$
192 比特	CRC $x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1$

所用的 CRC 的形式为  $g(x) = p(x) * (x+1)$ , 把已知的较好的原始多项式  $211_8(89_{16})$  和  $5361(AF116)$  用于  $p(x)$ 。

在现在技术中 CRC 检查所用的转换器 152 内的逻辑是众所周知的, 它由移位寄存器组成, 该移位寄存器是根据 CRC 多项式抽头的, 带有通过一异或(XOR)门累加的抽头输出端。

当在分组模式中对帧处理了四次或者在连续模式中对帧处理了一次, 而且四种速率的数据都已写入到 OB54 内时, 把线 156 上的中断设置成真(1)。SVD20 按序通过每个分组模式帧的相同的 432 个处理周期序列, 而与实际的数据速率或分组大小无关。因此, 线 156 上的中断发生在相对于线 64 上的输入码符号选通信号(图 7)固定的延时上。在连续模式中, 固定的延时与实际分组的大小有关。在七个内部 CHIPX8 时钟周期之后, 自动地把线 156 上的 INTD 中断复位。微处理器(未图示)能用这中断信号(或帧定时)来指示什么时候它应当读取输出数据。缓冲器 154 对于全部分组来说, 含有足够的存储空间, 因而, 允许通过微处理器接口 58(图 5)读取的输出数据的帧时间小于处理时间, 约为 10 毫秒。在分组模式时, 每个分组的最后八比特为零, 因为 SVD20 工作在子帧模式时假设对发送编码器进行预置, 并清洗至零状态。在每次读操作之后, 用缓冲器 154 来的下一个字节来自动地更新缓冲器 154 内的 DECDATA 寄存器(未图示)。数据是预取的, 以使微处理器等待下一字节的时间最少, 但预取仍需要至少四个内部 CHIPX8 时钟周期。如果微处理器能比每四个内部 XHIPX8 时钟周期快一倍地读 OB54, 那么, 微处理器必须测试 BYTERDY 总线 158 上的 BYTERDY 信号, 以保证同一输出字不能两

次被读到。微处理器能用包括在输出数据内的质量信息(“差错量度”),以上面引用的 Butler 等人的专利申请的方法从 OB54 中可获得的四个分组来选择最佳分组。

转换器 152 每 8 个处理周期向缓冲器 154 提供一个数据字节。在下一处理周期开始时把数据经内部总线 160 传送给缓冲器 154。当在线 56 上接收到数据的每位时,把数据计时入转换器 152 内的 CRC 产生器(未图示)内。在每个分组开始时把 CRC 产生器复位到全 1。在 96 和 192 比特分组的结束处,检查 CRC 的余数,如果余数为全零,则设置 STATUS 寄存器(未图示)内的 CRC 位。在发送 INTD 线 156 上的中断之前,在每次读操作之后,预取读数据。当在线 156 上发送 INTD 中断时,把读地址指针设置到零地址。在每次读操作之后,把线 158 上的 BYTERDY 信号设置成零并保持,一直到有新的数据字节被锁存到缓冲器 154 内的 DECDATA 寄存器(未图示)内。OB54 还包含测试逻辑,可以对固有的功能进行自测试。

#### 本发明的控制器和处理器接口部件

参见图 5,控制器 62 为 SVD20 提供定时和控制选通信号。SVD20 以内部 CHIPX8 时钟速率运行,控制器 62 根据该内部时钟速率提供一串选通信号和启动信号。这些控制信号能通过用于测试目的的复位操作来进行初始准备,但在正常操作时,它们响应于总线 68 上的 DECMODE 信号按序通过一组固定的操作,然后返回到空闲状态。每次在线 70(图 7)上出现译码器同步选通信号之后,控制信号序列再次开始。指示有效的反向链接数据的链接有效信号工作,在线 150 启动 RSTB 选通信号,它使数据开始传送给 OB54。控制器 62 和微处理器接口 58 包含固有的操作所必需的微处理器控制寄存器,包括引脚控制和复位寄存器。在这两个部件中还包括有专门的自测试逻辑。

上述描述的 SVD20 的每个部件的内部模块可以用该技术领域

中已知的实用的方式来实现,但最好实现成单片式集成电路芯片。

虽然本描述中的讲授、举例和实施都假设为串行处理来实现所希望的目的,但对于本技术领域的熟练人员来说很显然,译码器逻辑可以重现为多路并行译码器,它同时工作以提供如传输系统内的数据速率一样多的输出。

本发明还可以变化,例如对如系统要求一样多的数据速率使用循环冗余编码。

显然,对于本技术领域的普通人员来说通过这些讲述,能容易地得出本发明的其它实施例和变化。因此,本发明应仅受下面的权利要求书所限制,它包括了所有的结合上面的说明和附图所考虑的这些实施例和变化。

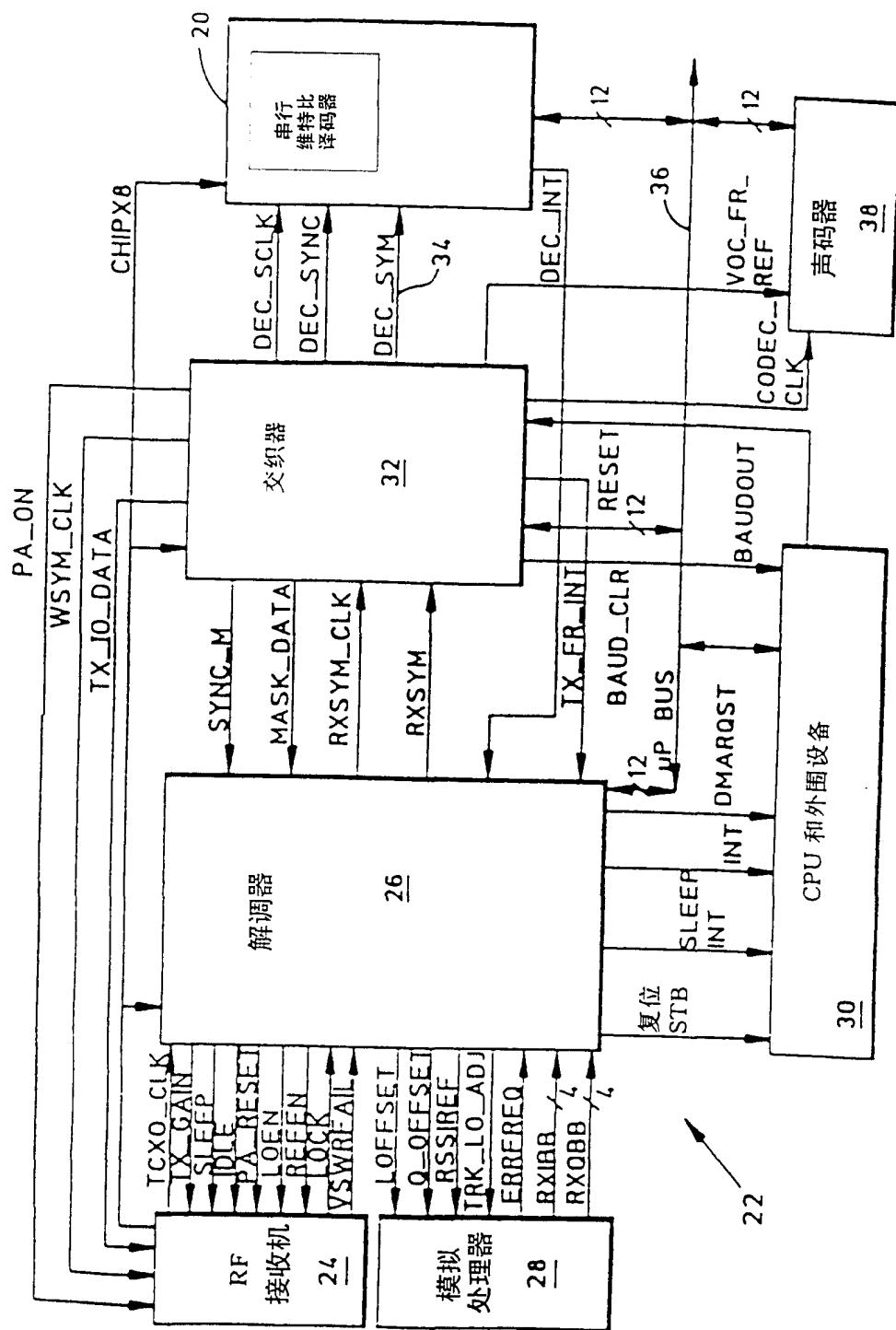


图 1

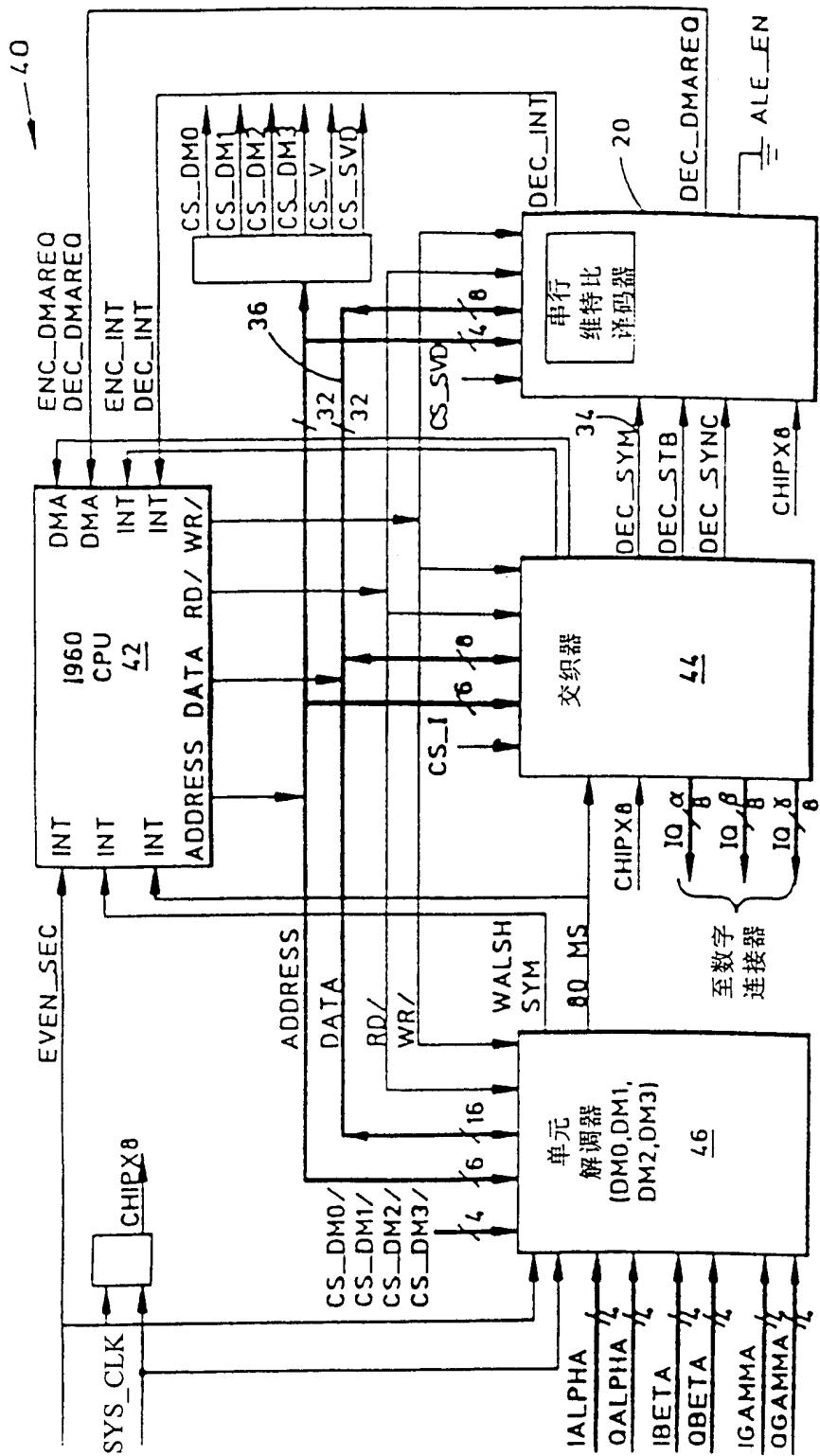


图 2

参数	数据速率(bps)		单位
	1200		
PN 片码速率	1.2288		Mcps
码速率	1/2		位/码符号
码重复	2		调制符号/码符号 *
调制符号速率	4,800		sps
PN 片码/调制符号	256		PN 片码/调制符号
PN 片码/位	1024		PN 片码/位

\* 一个码符号的每次重复是一个调制符号

图 3A

参数	数据速率(bps)			单位
	9600	4800	2400	
PN 片码速率	1.2288	1.2288	1.2288	Mcps
码速率	1/2	1/2	1/2	位/码符号
码重复	1	2	4	调制符号/码符号 *
调制符号速率	19,200	19,200	19,200	sps
PN 片码/调制符号	64	64	64	PN 片码/调制符号
PN 片码/位	128	256	256	PN 片码/位

\* 一个码符号的每次重复是一个调制符号

图 3B

参数	数据速率(bps)				单位
	9600	4800	2400	1200	
PN 片码速率	1.2288	1.2288	1.2288	1.2288	Mcps
码速率	1/2	1/2	1/2	1/2	位/码符号
码重复	1	2	4	8	调制符号/码符号 *
调制符号速率	19,200	19,200	19,200	19,200	sps
PN 片码/调制符号	64	64	64	64	PN 片码/调制符号
PN 片码/位	128	256	512	1024	PN 片码/位

\* 一个码符号的每次重复是一个调制符号

图 3C

参数	数据速率(bps)				单位
	9600	4800	2400	1200	
PN 片码速率	1.2288	1.2288	1.2288	1.2288	Mcps
码速率	1/3	1/3	1/3	1/3	位/码符号
发射机占空度	100.0	50.0	25.0	12.5	%
码符号速率	28,800	28,800	28,800	28,800	sps
调制	6	6	6	6	码符号/Walsh 符号
Walsh 符号率	4800	4800	4800	4800	sps
Walsh 片码率	307.20	307.20	307.20	307.20	kcps
Walsh 符号	208.33	208.33	208.33	208.33	μs
PN 片码/码符号	42.67	42.67	42.67	42.67	PN 片码/码符号
PN 片码/Walsh 符号	256	256	256	256	PN 片码/Walsh 符号
PN 片码/Walsh 片码	4	4	4	4	PN 片码/Walsh 片码

图 3D

参数	数据速率(bps)		单位
	4800	1.2288	
PN 片码速率		Mcps	
码速率	1/3	位/码符号	
发射机占空度	100.0	%	
码符号速率	28,800	sps	
调制	6	码符号/Walsh 符号	
Walsh 符号率	4800	sps	
Walsh 片码率	307.20	kcps	
Walsh 符号	208.33	μs	
PN 片码/码符号	42.67	PN 片码/码符号	
PN 片码/Walsh 符号	256	PN 片码/Walsh 符号	
PN 片码/Walsh 片码	4	PN 片码/Walsh 片码	

图 3E

控制 信号	正向			反向	
	同步	呼叫	通话	通话	接入
码符号调制速率	4,800	19,200	19,200	28,800	28,800
原始位数据速率	1200	固定	1200-9600	1200-9600	4800
卷积码速率	1/2	1/2	1/2	1/3	1/3
符号重复	2	1.2 或 4	N/A	N/A	2
编码模式	连续	连续	分组	分组	分组
传输模式	重复	重复	重复	单个分帧	重复
在符号#时启动译码	2	2	2	8	8
图	3A	3B	3C	3D	3E

图 4

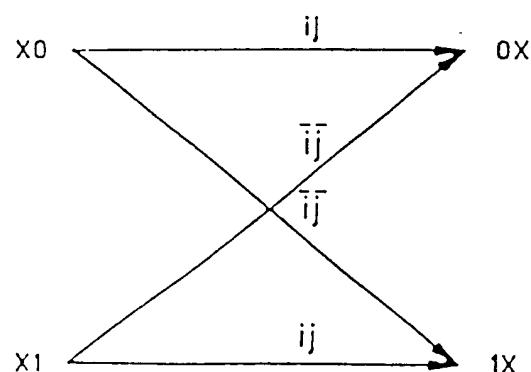


图 6B

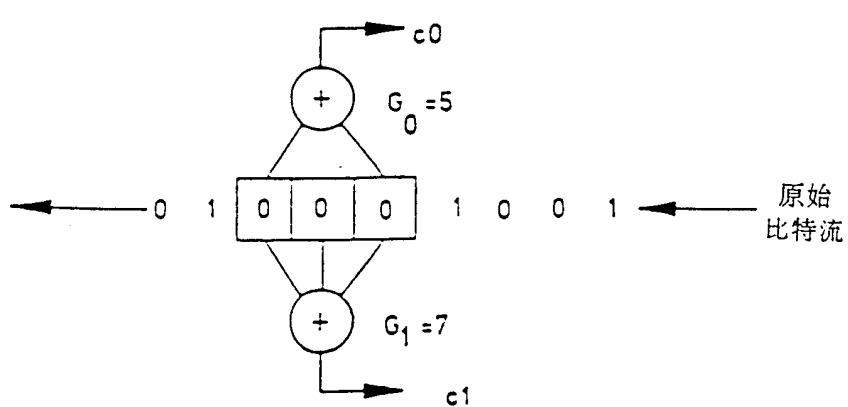
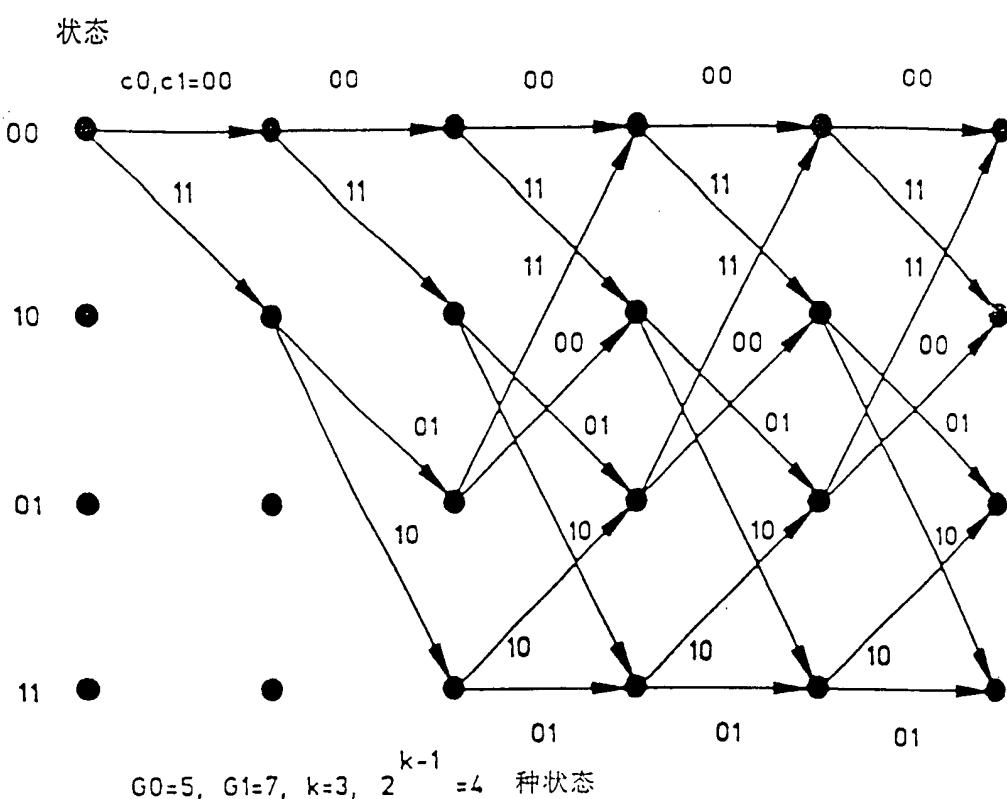
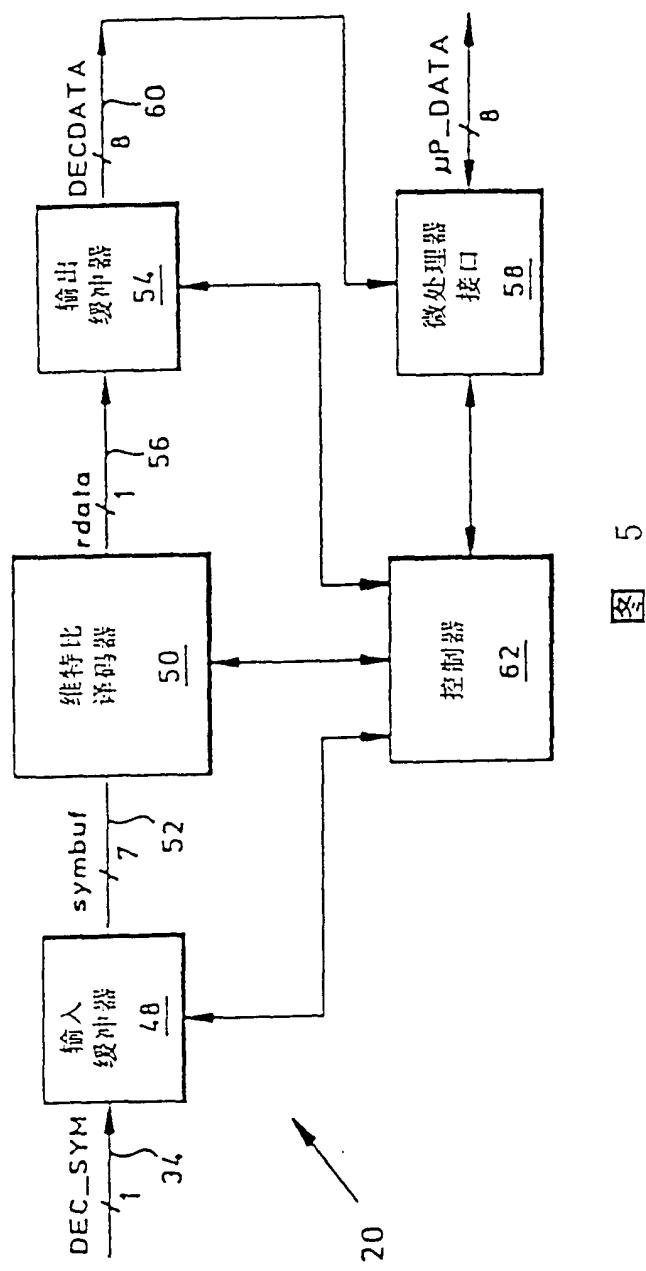


图 6A



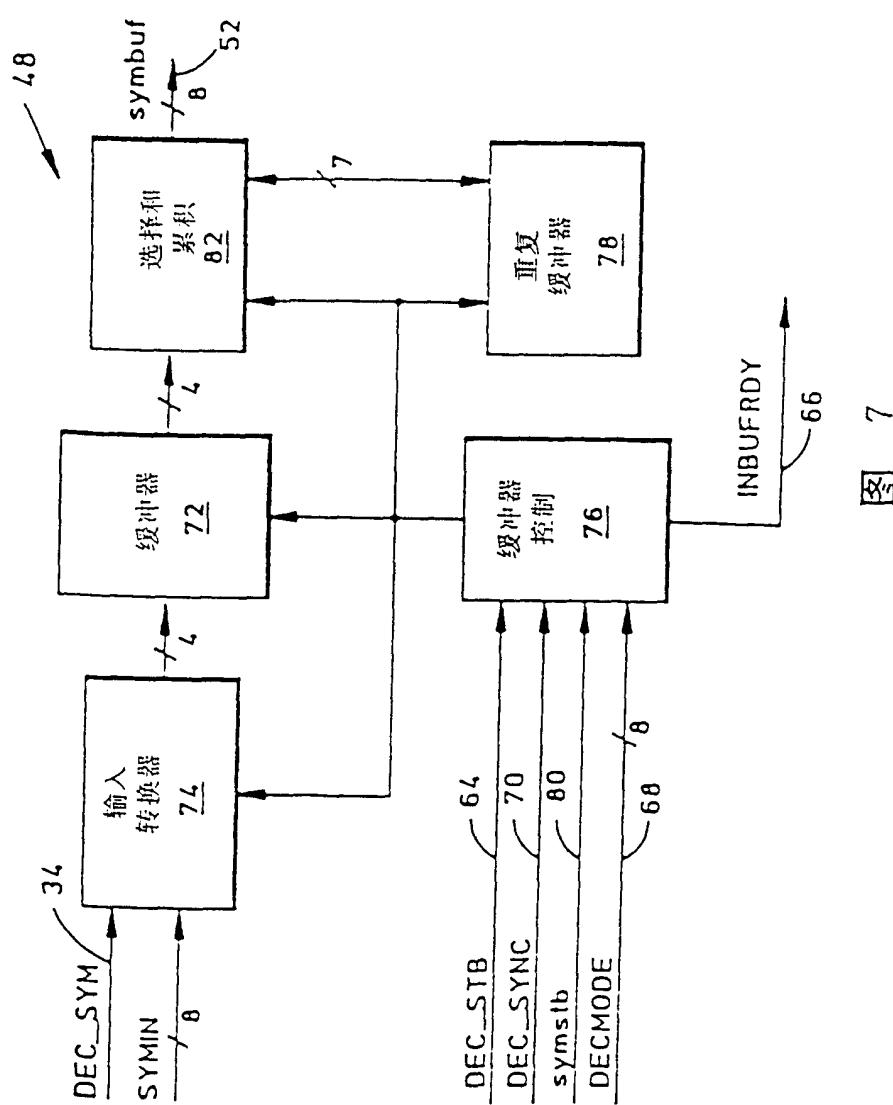


图 7

DBR 速率	DBR 位	选择出的行		
		DBR 位值 ->	0	1
1/2	DBR0	0,1	2,3	
1/2	DBR1	4,5	6,7	
1/2	DBR2	8,9	10,11	
1/2	DBR3	12,13	14,15	
1/2	DBR4	16,17	18,19	
1/2	DBR5	20,21	22,23	
1/2	DBR6	24,25	26,27	
1/2	DBR7	28,29	30,31	
1/4	DBR8	DBR0	DBR1	
1/4	DBR9	DBR2	DBR3	
1/4	DBR10	DBR4	DBR5	
1/4	DBR11	DBR6	DBR7	
1/8	DBR12	DBR8	DBR9	
1/8	DBR13	DBR10	DBR11	

图 8

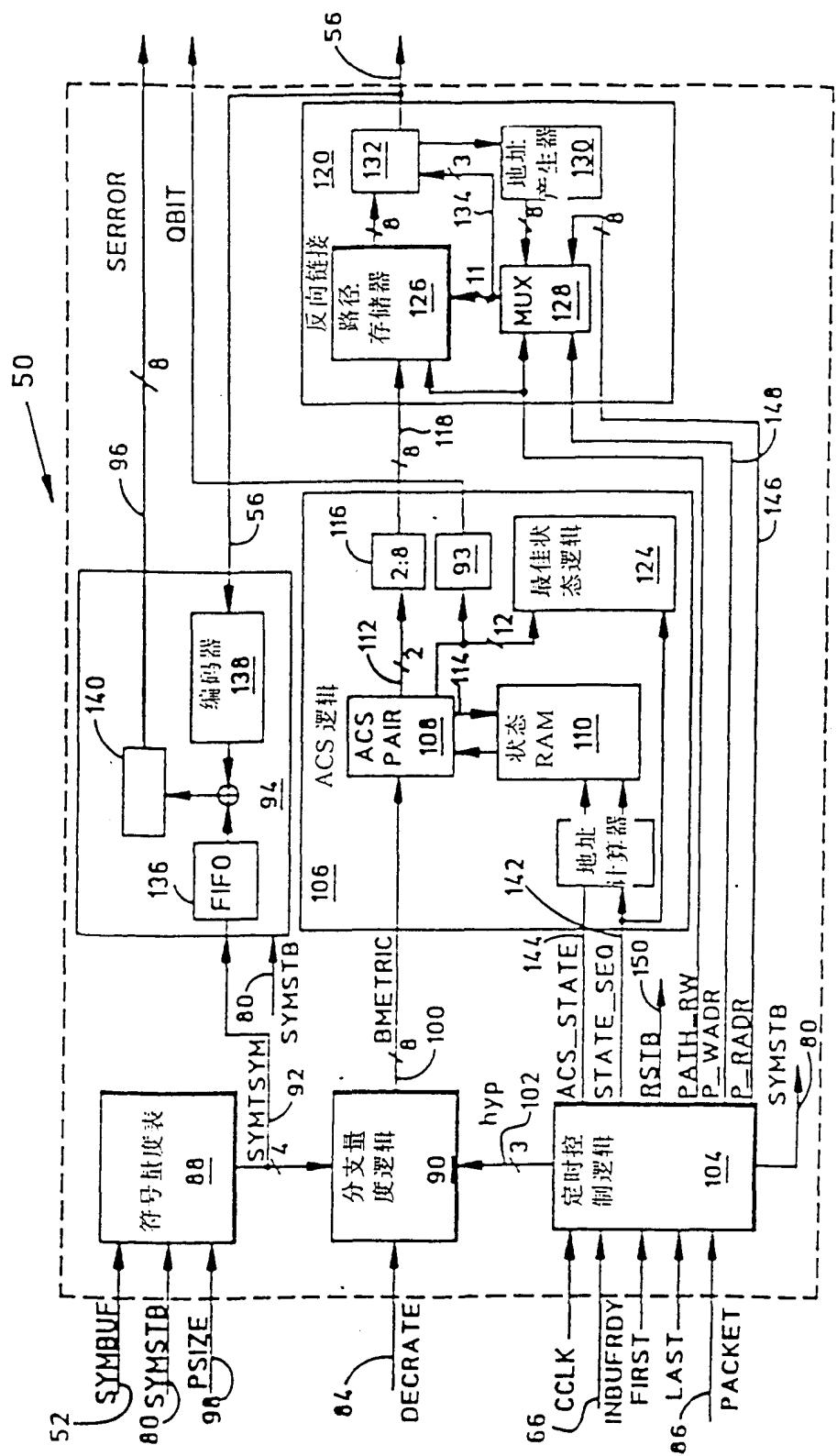


图 9

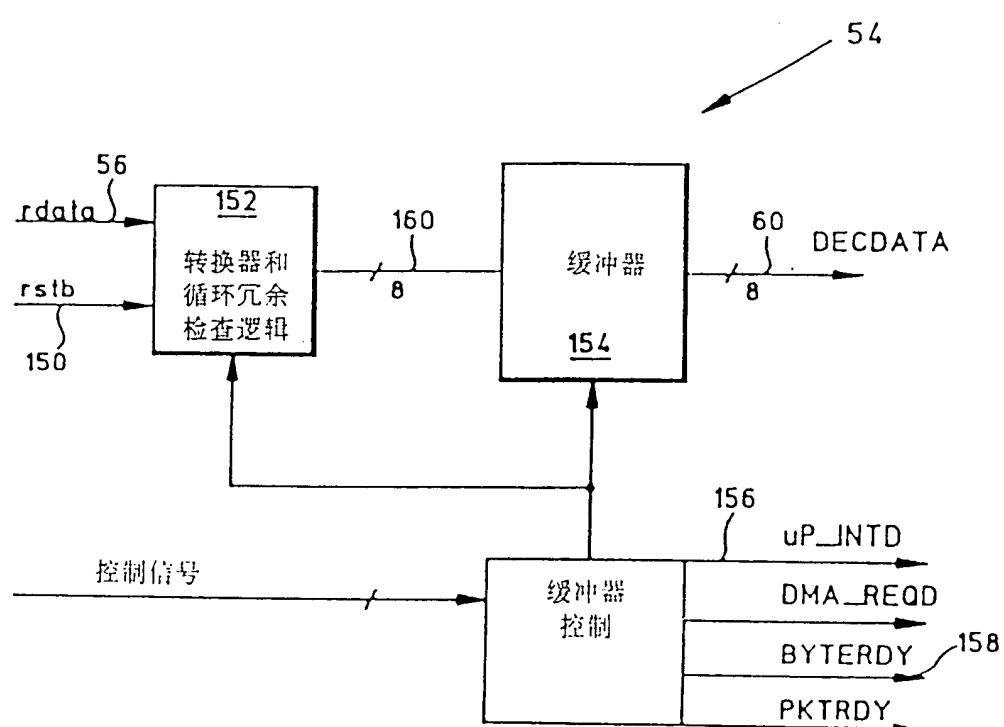


图 10

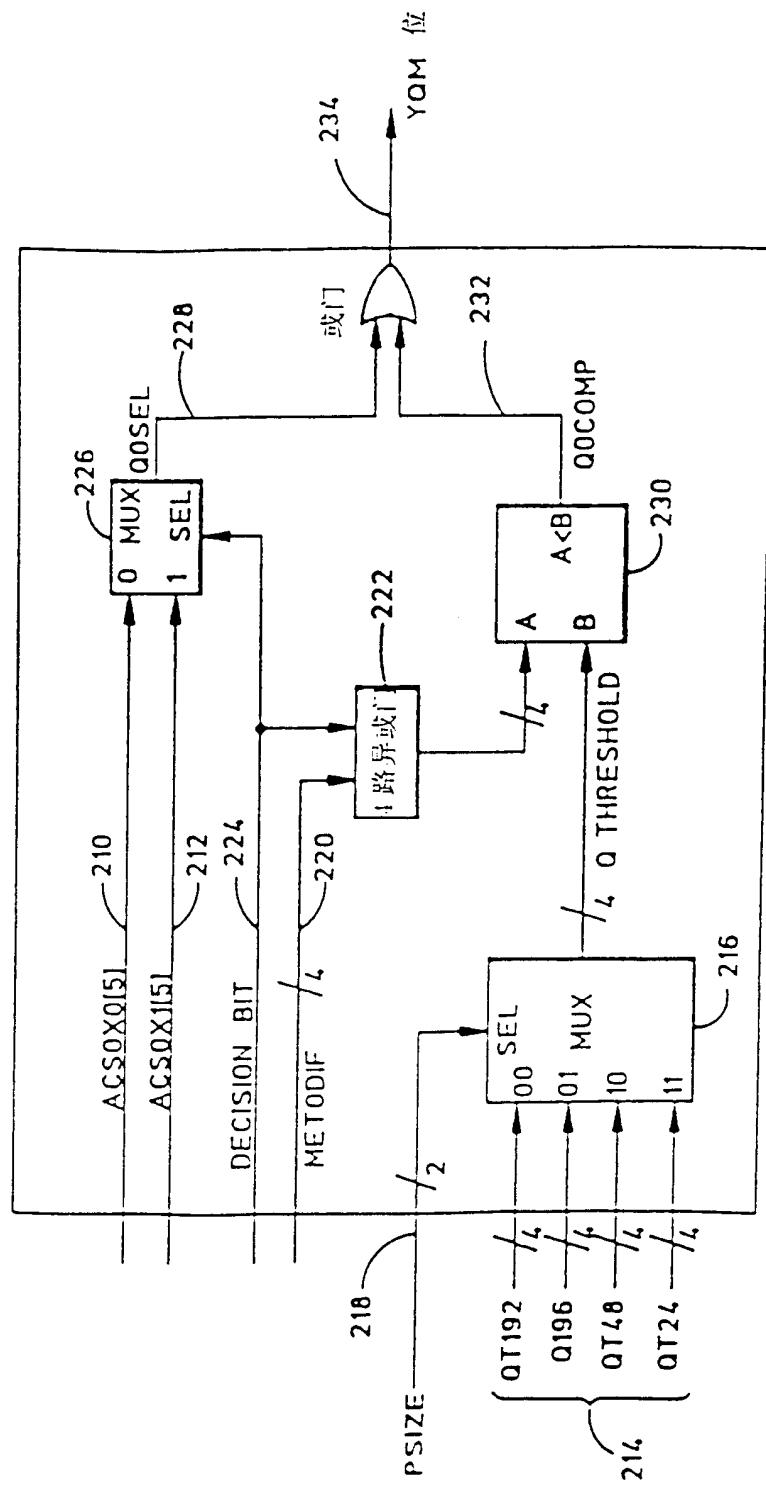


图 11