



(12) 发明专利

(10) 授权公告号 CN 101350709 B

(45) 授权公告日 2011. 01. 05

(21) 申请号 200710137821. X

CN 1210423 A, 1999. 03. 10, 全文.

(22) 申请日 2007. 07. 20

审查员 苏宁

(73) 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

(72) 发明人 梁伟光 耿东玉 封东宁

(74) 专利代理机构 上海明成云知识产权代理有
限公司 31232

代理人 成春荣 竺云

(51) Int. Cl.

H04L 7/02 (2006. 01)

(56) 对比文件

EP 1515457 A2, 2005. 03. 16, 全文.

CN 1859047 A, 2006. 11. 08, 全文.

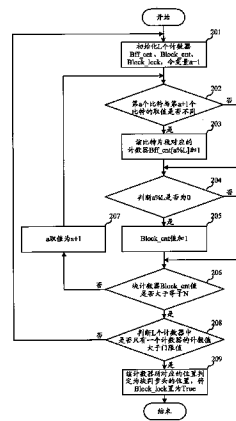
权利要求书 2 页 说明书 11 页 附图 9 页

(54) 发明名称

块、帧同步方法及装置

(57) 摘要

本发明涉及通信领域,公开了一种块、帧同步方法及装置,在输入误码率比较高时,能够更准确地完成同步。本发明中,检测 N 个比特组,对每一个比特组中各个位置出现与块同步头相同的比特片段的次数进行统计,如果只有一个位置的统计值大于第一门限,则该位置即为块同步头的位置。对由 M 个块同步头所组成的接收序列进行 M-1 次循环位移,可以得到接收序列各种可能的循环位移结果,将这些循环位移结果与参考序列进行相关运算,可以用相关值定量地指示各种循环位移结果与参考序列接近的程度,通过将 K 个比特组中对应相同循环移次数的相关值相加,能够可靠地知道哪一种循环位移结果与参考序列最为接近,从而准确知道帧头的位置。



CN 101350709 B

1. 一种块同步方法,其特征在于,包括以下步骤:

初始化 L 个计数器,每个计数器分别对应于在长度为 L 的一个比特组中的一个位置,块同步头出现在所述 L 个计数器所对应的 L 个位置之一, $L > 1$;

对 N 个比特组进行检测,每个比特组长度为 L,如果检测到与预设的块同步头之一相同的比特片段,则在所述 L 个计数器中选择该比特片段所在位置对应的计数器,增加所选择的计数器的计数值; $N > 1$;

如果所述 L 个计数器中只有一个计数器的计数值大于第一门限,则该计数器所对应的位置判定为块同步头的位置;其中所述块的长度为 L 比特。

2. 根据权利要求 1 所述的块同步方法,其特征在于,所述块同步头为 2 比特,所述预设的块同步头为“10”或“01”。

3. 根据权利要求 2 所述的块同步方法,其特征在于,所述对 N 个比特组进行检测的步骤包括以下子步骤:

对每个比特组中的每个比特进行扫描,如果被扫描的比特组中第 X 个比特与第 X+1 个比特的取值不同,则判定为检测到与预设的块同步头之一相同的比特片段,所检测到的比特片段由所述第 X 个比特与第 X+1 个比特组成;

所述比特片段所在位置对应的计数器增加计数值的步骤包括以下子步骤:

将第 X 个所述计数器的计数值加 1。

4. 根据权利要求 1 所述的块同步方法,其特征在于,在所述对 N 个比特组进行检测的步骤之后,还包括以下步骤:

如果所述 L 个计数器的计数值都小于或等于所述第一门限,或者所述 L 个计数器中有至少两个计数器的计数值大于所述第一门限,则重新进行块同步。

5. 根据权利要求 1 所述的块同步方法,其特征在于,在所述对 N 个比特组进行检测的步骤中,还包括以下子步骤:

逐个对所述比特组进行检测,在完成对所述 N 个比特组的检测之前,如果所述 L 个计数器中有至少两个计数器的计数值大于所述第一门限,则立即停止当前的检测,重新进行块同步。

6. 根据权利要求 1 至 5 中任一项所述的块同步方法,其特征在于,所述 N 为每个帧中块的数目;

所述帧中各个块的块同步头均为所述预设的块同步头之一,所述第一门限 = N- 块偏置值;

或者,

在所述帧中 N-P 个块的块同步头为所述预设的块同步头之一, P 个块的块同步头不属于所述预设的块同步头之一,所述第一门限 = N- 块偏置值 -P。

7. 根据权利要求 6 所述的块同步方法,其特征在于,所述块偏置值根据所述块中预设的误码率确定。

8. 一种块同步装置,其特征在于,包括:

L 个计数器,每个计数器分别对应于在长度为 L 的一个比特组中的一个位置,块同步头出现在所述 L 个计数器所对应的 L 个位置之一, $L > 1$;

检测模块,用于对 N 个比特组进行检测,每个比特组长度为 L,如果检测到与预设的块

同步头之一相同的比特片段,则在所述 L 个计数器中选择该比特片段所在位置对应的计数器,指示所选择的计数器增加计数值;

初始化模块,用于在所述检测模块进行检测前对所述 L 个计数器进行初始化;

块同步判决器,用于在所述检测模块完成检测后判断所述 L 个计数器中是否只有一个计数器的计数值大于第一门限,如果是,则将该计数器所对应的位置确定为块同步头的位置;

其中所述块的长度为 L 比特, $N > 1$ 。

9. 根据权利要求 8 所述的块同步装置,其特征在于,所述块同步头为 2 比特,所述预设的块同步头为“10”或“01”。

10. 根据权利要求 9 所述的块同步装置,其特征在于,所述检测模块包括:

移位寄存器,用于保存当前被检测的两个比特;

比较器,用于比较所述移位寄存器中的两个比特,如果所述两个比特不同,则指示所述两个比特所组成的比特片段所对应的计数器增加计数值。

11. 根据权利要求 8 至 10 中任一项所述的块同步装置,其特征在于,所述块同步判决器判定所述 L 个计数器的计数值都小于或等于所述第一门限,或者所述 L 个计数器中有至少两个计数器的计数值大于所述第一门限,则指示所述初始化模块对所述 L 个计数器重新进行初始化,再指示所述检测模块对新的 N 个比特组进行检测。

12. 根据权利要求 8 至 10 中任一项所述的块同步装置,其特征在于,还包括:

检测中止模块,用于在所述检测模块对 N 个比特组的检测过程中,判断所述 L 个计数器中是否有至少两个计数器的计数值大于所述第一门限,如果是,则指示所述初始化模块对所述 L 个计数器重新进行初始化,再指示所述检测模块对新的 N 个比特组进行检测。

块、帧同步方法及装置

技术领域

[0001] 本发明涉及通信领域,特别涉及块和帧同步技术。

背景技术

[0002] 以太网无源光网络 (Ethernet Passive Optical Network,简称“EPON”) 接入技术是一种比较好的接入技术,其主要优点是维护简单、成本较低、较高的传输带宽和高性价比。

[0003] 但是,由于 EPON 是一种采用无源光传输的技术,在这个网络中不使用具有放大和中继功能的元器件。因此 EPON 网络的传输距离和分支数目依赖于功率预算和各种传输损耗。随着传输距离或分支数目的增加,传输数据的信噪比 (Signal Noise Ratio,简称“SNR”) 逐渐减小,传输的数据的误码率逐渐增高。为了解决这一问题,在 EPON 系统中引入了前向纠错 (ForwardError Correction,简称“FEC”) 技术来提高系统的抗干扰能力,从而增大系统的功率预算。

[0004] 然而,64b/66b 和 64b/65b 是两种编码效率较高的线路编码机制,使用了带有非扰码同步字符和控制字符的扰码方式。在 EPON 系统物理编码子层 (Physical Coding Sublayer,简称“PCS”) 中,使用了线路编码技术来提高数据传输的正确率。

[0005] 目前 FEC 信息帧结构方案如图 1 所示,进入 PCS 层的数据先经过 64b/66b 线路编码,形成以 66 比特为单位的线路编码块。64b/66b 编码机制是在 64 比特数据信息的基础上,增加了 2 比特的同步头。这 2 比特的同步头只有“01”或“10”这两种可能。其中,同步头为“01”表示 64 比特全部为数据信息;同步头为“10”表示 64 比特信息中包含数据信息和控制信息;而同步头为“00”或“11”则表示传输过程中发生了错误。在接收端,需要 64b/66b 线路编码中的两比特“01”或“10”的同步头实现 64b/66b 线路编码块同步 (blocksynchronization),以便于根据同步头来判断线路编码块中数据的类型,然后实现信息帧同步 (frame synchronization),以便于后续对 FEC 的解码。

[0006] 现有技术的块同步和 FEC 码字信息帧同步的实现原理分别如下:

[0007] 块同步原理为:在接收端,设置一个长度为 66 比特的缓存寄存器,在同步开始之前,将缓存寄存器中的所有位置进行初始化设置为“True(真)”。然后将所输入的比特数据与其前面的比特数据进行比较,如果这 2 比特数据相同,则就缓存寄存器中相应的位置置为“false(假)”。由于 64b/66b 编码块同步头的 2 比特数据只会是互异的,则当输入一定数目的数据后,缓存寄存器的某个的位置的值一直为“True”,那么这个位置就确定为 64b/66b 编码块的同步头起始的位置,从而实现块同步。

[0008] 而 FEC 信息帧同步过程的机理和块同步的原理类似,区别在于要把缓存寄存器的长度增加到一个 FEC 信息帧的长度。在进行同步前将此缓存寄存器所有位置置为“True”,将输入的比特数据与其前面的比特数据进行比较,利用 64b/66b 编码块同步头互异的特性以及校验块同步头相同的特性,可以确定该缓存寄存器中数值一直为“True”的位置是同步头的位置,从而实现 FEC 信息帧同步。

[0009] 另外,还有一种比较简单的FEC信息帧同步方法,该方法只对FEC帧中的校验信息块同步头进行扫描,由于校验信息块的2比特同步头永远是相同的,在缓存寄存器的对应位置上的值会一直是真,这样确定校验信息块的开始位置,从确定FEC帧的起始位置,实现FEC帧同步。

[0010] 如果其中一个64b/66b线路编码块或FEC校验信息块的同步头发生错误,缓存寄存器中可能不会出现“True”的状态,此时就无法完成块同步或FEC帧同步。一般采取重新将缓存寄存器初始化,然后重新进行同步操作来解决。然而,本发明的发明人发现,在信道中的干扰较强时,接收数据的误码率较高,这样同步头错误的几率也很高,使得无法准确的完成同步甚至无法实现同步。另外,由于同步头错误时,只能通过重新初始化后进行同步操作,这样就会增长时延,影响系统的性能。

发明内容

[0011] 本发明实施方式要解决的主要技术问题是提供一种块、帧同步方法及装置,使得在输入误码率比较高时,能够更准确地完成同步。

[0012] 为解决上述技术问题,本发明的实施方式提供了一种块同步方法,包括以下步骤:

[0013] 初始化L个计数器,每个计数器分别对应应在长度为L的一个比特组中的一个位置,块同步头出现在L个计数器所对应的L个位置之一, $L > 1$;

[0014] 对N个比特组进行检测,每个比特组长度为L,如果检测到与预设的块同步头之一相同的比特片段,则在L个计数器中选择该比特片段所在位置对应的计数器,增加所选择的计数器的计数值; $N > 1$;

[0015] 如果L个计数器中只有一个计数器的计数值大于第一门限,则该计数器所对应的位置判定为块同步头的位置;其中块的长度为L比特。

[0016] 本发明的实施方式还提供了一种块同步装置,包括:

[0017] L个计数器,每个计数器分别对应应在长度为L的一个比特组中的一个位置,块同步头出现在L个计数器所对应的L个位置之一, $L > 1$;

[0018] 检测模块,用于对N个比特组进行检测,每个比特组长度为L,如果检测到与预设的块同步头之一相同的比特片段,则在L个计数器中选择该比特片段所在位置对应的计数器,指示所选择的计数器增加计数值;

[0019] 初始化模块,用于在检测模块进行检测前对L个计数器进行初始化;

[0020] 块同步判决器,用于在检测模块完成检测后判断L个计数器中是否只有一个计数器的计数值大于第一门限,如果是,则将该计数器所对应的位置确定为块同步头的位置;

[0021] 其中块的长度为L比特, $N > 1$ 。

[0022] 本发明的实施方式还提供了一种帧同步方法,每个帧由M个块组成, $M > 1$,包括以下步骤:

[0023] 对K个比特组进行检测,每个比特组包括M个块,其中,对每个比特组进行检测时,读取该比特组内M个块的块同步头组成接收序列,以块同步头的长度为单位对接收序列进行所有可能的循环移位,将该接收序列经循环移位所有可能得到的排列与参考序列进行相关运算,记录每次相关运算所得的相关值, $K > 1$;

- [0024] 将 K 个比特组中对应相同循环移次数的相关值相加,得到 M 个相关和;
- [0025] 如果只有一个相关和大于第二门限,则根据该相关和所对应的循环移次数得到帧头的位置。
- [0026] 本发明的实施方式还提供了一种帧同步装置,包括:
- [0027] 检测单元,用于对 K 个比特组进行检测,每个比特组包括 M 个块,其中,对每个比特组进行检测时,读取该比特组内 M 个块的块同步头组成接收序列,以块同步头的长度为单位对接收序列进行所有可能的循环移位,将该接收序列经循环移位所有可能得到的排列与参考序列进行相关运算,记录每次相关运算所得的相关值, $K > 1, M > 1$;
- [0028] 累加器,用于将检测单元得到的 K 个比特组中对应相同循环移次数的相关值相加,得到 M 个相关和;
- [0029] M 个寄存器,分别用于存储累加器得到的 M 个相关和;
- [0030] 帧同步判断器,用于判断 M 个寄存器中是否只有一个相关和大于第二门限,如果是,则根据该相关和所对应的循环移次数得到帧头的位置。
- [0031] 本发明实施方式与现有技术相比,主要区别及其效果在于:
- [0032] 检测 N 个比特组,对每一个比特组中各个位置出现与块同步头相同的比特片段的次数进行统计,如果只有一个位置的统计值大于第一门限,则该位置即为块同步头的位置。在输入误码率比较高时只是使各个位置的统计值较为接近,但块同步头位置的统计值明显大于其它位置的统计值的概率依然很大,因此仍然能够准确地完成同步。

附图说明

- [0033] 图 1 是现有技术中 FEC 信息帧结构示意图;
- [0034] 图 2 是根据本发明第一实施方式的块同步方法流程图;
- [0035] 图 3 是根据本发明第二实施方式的块同步装置结构示意图;
- [0036] 图 4 是根据本发明第三实施方式的块同步方法流程图;
- [0037] 图 5 是根据本发明第三实施方式的进行块同步的 FEC 信息帧结构示意图;
- [0038] 图 6 是根据本发明第三实施方式的块同步方法流程图;
- [0039] 图 7 是根据本发明第四实施方式的块同步装置结构示意图;
- [0040] 图 8 是根据本发明第五实施方式的帧同步方法的原理示意图;
- [0041] 图 9 是根据本发明第五实施方式的帧同步方法流程图;
- [0042] 图 10 是根据本发明第五实施方式的进行帧同步的同步相关运算原理示意图;
- [0043] 图 11 是根据本发明第六实施方式的帧同步装置结构示意图。

具体实施方式

[0044] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明的实施方式作进一步地详细描述。

[0045] 本发明第一实施方式涉及一种块同步方法,所接收的数据流中各个块的块同步头为 2 比特,预设的块同步头为“10”或“01”。其中,块的长度为 L 比特,例如,块的长度为 66 比特(即 $L = 66$),在系统中相应地设置 L 个计数器,也即设置 66 个计数器。在进行块同步时,以长度 L 比特为一个比特组,对 N 个比特组的数据流进行检测,其中, L 个计数器分别对

应在一个比特组中一个可能的块同步头位置。该块同步方法的具体流程如图 2 所示。

[0046] 在步骤 201 中,在同步开始时,初始化 $L(L = 66)$ 个计数器,也即将 66 个计数器 $Bff_cnt[a \% L]$ 全部清零,其中,令 $a = L$ 时, $Bff_cnt[0]$ 指向第 L 个计数器,同时将块计数器 $Block_cnt$ 置零,将表示同步状态的变量 $Block_lock$ 置为“false”,也即“ $Block_lock \leq false$ ”。其中, $a \% L$ 表示 a 对 L 取模,当 a 小于等于 L 时, $a \% L = a$;当 a 大于 L 时,则 $a \% L$ 等于 L 除以 a 后所得的余数。

[0047] 在步骤 202 中,对 N 个比特组中的每个比特进行扫描检测,在本实施方式中,预设的块同步头为“10”或“01”,因此只需判断被扫描的比特组中第 a 个比特与第 $a+1$ 个比特的取值是否不同,即对每个比特组执行 $if(input[a] \neq input[a+1])$ 语句,其中,函数 $input[a]$ 表示读取输入的数据流中 $Block_cnt$ 所指示的比特组中的第 a 个比特的数据,因此该语句表示判断当前 $Block_cnt$ 指示的比特组中的第 a 个比特的数据和第 $a+1$ 个比特的数据是否互异,通过简单的判断两个比特数据是否互异,可以快速地判断两个相邻比特是否为预设的块同步头之一。如果所判断的两个比特数据的取值不同,则 $if(input[a] \neq input[a+1])$ 语句的值为“true”,进入步骤 203,将该比特片段所在位置对应的计数器 $Bff_cnt[a \% L]$ 的计数值加 1,然后转入步骤 204;否则, $if(input[a] \neq input[a+1])$ 语句的值为“false”,计数器 $Bff_cnt[a \% L]$ 不做动作,也即计数值不变,直接转入步骤 204。

[0048] 在步骤 204 中,判断 $a \% L$ 是否为 0,只有当 $a = L$ 时, $a \% L$ 的值为 0,如果 $a \% L$ 的值为 0,则表示本比特组中的每一个比特分别同其相邻比特之间进行了比较,进入步骤 205,块计数器 $Block_cnt$ 加 1,使得可以对下一个比特组的数据进行扫描,然后进入步骤 206;如果 $a \% L$ 的值不为 0,则表示本比特组中每一个比特与其相邻比特之间的比较还未完成,块计数器 $Block_cnt$ 值不变,将继续对该比特组进行扫描,并直接进入步骤 206。

[0049] 在步骤 206 中,判断块计数器的 $Block_cnt$ 值是否大于等于 N ,如果 $Block_cnt$ 值小于 N ,进入步骤 207,将 a 取值为“ $a+1$ ”,然后转入步骤 202;如果 $Block_cnt$ 值大于等于 N ,则表示对 N 个比特组扫描完毕,转入步骤 208。

[0050] 在步骤 208 中,判断 L 个计数器中是否只有一个计数器的计数值大于块门限值 ($B_threshold$),其中, $B_threshold = N -$ 块偏置值 (B_offset), B_offset 可以根据块预设的误码率确定,预设的误码率越高, B_offset 值越大。通过灵活设置 B_offset 可以适应不同的输入误码率。

[0051] 如果 L 个计数器的计数值都小于或等于 $B_threshold$,或者 L 个计数器中有至少两个计数器的计数值大于 $B_threshold$,那么根据 $Block_lock$ 值为“false”,返回步骤 201,重新进行块同步操作对后续数据流进行同步检测;如果 L 个计数器中只有一个计数器的计数值大于 $B_threshold$,则进入步骤 209,执行“ $Block_lock \leq true$ ”语句,将变量 $Block_lock$ 置为“true”,判定块同步完成,该计数器所对应的位置判定为块同步头的位置。由于在该过程中利用计数器实现统计,在输入误码率比较高时只是使各个位置的统计值较为接近,但块同步头位置的统计值明显大于其它位置的统计值的概率依然很大,因此,通过本实施方式仍然能够准确地完成同步。

[0052] 本实施方式的检测和判断的具体过程如下:

[0053] 对输入的第 1 组的 66 个比特的数据进行检测分析,具体检测分析过程为:将该组的第 2 个比特的数值同第 1 个比特的数值进行比较,如果它们的取值不同,为“10”或“01”,

则表示由第 1 组第 1 个比特和第 1 组第 2 个比特所组成的比特片段与预设的块同步头相同,那么,将第 1 个计数器加 1;如果它们的取值相同,为“00”或“11”,则表示由第 1 组第 1 个比特和第 1 组第 2 个比特所组成的比特片段与预设的块同步头不同,那么,保持第 1 个计数器的值不变。然后,依次将该组的第 3 个比特的数值同第 2 个比特的数值进行比较,如果它们的取值不同,则第 2 个计数器加 1;如果它们的取值相同,则第 2 个计数器的值不变。依次类推,对第 1 组中的 66 个比特的数据依次进行检测分析。

[0054] 当 $a = 66$ 时, $a+1$ 则为 67,将收到数据的第 66 个比特和第 67 个比特进行比较,如果它们的取值不同,则第 66 个计数器加 1($a = L$ 时,令 $Bff_cnt[a \% L]$ 即 $Bff_cnt[0]$ 指向第 L 个计数器);如果它们的取值相同,则第 66 个计数器的值不变。然后,当 $a = 67$, $a+1 = 68$ 时,对第 67 个比特和第 68 个比特的数据进行比较,如果它们的取值不同,则 $Bff_cnt[a \% L]$ 所指向的第 1 个计数器的值加 1;如果它们的取值相同,则第 1 个计数器的值保持不变。容易理解,对于 N 个比特组的 $66 \times N$ 个比特数据的扫描分析只要采取上述的方法依次进行即可。

[0055] 通过实验证明:当 N 的取值为 30, B_offset 的取值为 4,在信道编码的误比特率(Bit Error Rate,简称“BER”)为 10^{-3} 时,采用本实施方式可以使得正确同步的概率达 99.8%,而采用现有技术方案的正确同步的概率为 94.2%;当 N 的取值为 60, B_offset 的取值为 5,信道编码 BER 为 10^{-3} 时,采用本实施方式可以使得正确同步的概率接近 100%,而采用现有技术方案的正确同步的概率为 88.7%;当 N 的取值为 30, B_offset 的取值为 4,信道编码的 BER 为 10^{-2} 时,采用本实施方式可以使得正确同步的概率为 99.8%,而采用现有技术方案的正确同步的概率为 54.8%;当 N 的取值为 60, B_offset 的取值为 5,信道编码的 BER 为 10^{-2} 时,采用本实施方式可以使得正确同步的概率为 99.98%,而采用现有技术方案的正确同步的概率为 30.1%。显而易见,当信道中干扰比较大时,现有技术方案正确同步的概率很低,甚至不能同步,而采用本发明的实施方式可以维持较高的正确同步率,可以使得系统的性能大大地提高,显示了本发明方案的优越性。

[0056] 本发明第二实施方式涉及一种块同步装置,如图 3 所示,包含 L (例如, $L = 66$) 个计数器、检测模块、初始化模块和块同步判决器。其中, L 为块的长度,单位为比特。块同步头为 2 比特,预设的块同步头为“10”或“01”。

[0057] 具体地说,66 个计数器,每个计数器分别对应于在长度为 66 比特的一个比特组中一个可能的块同步头位置;检测模块,用于对 $N(N > 1)$ 个比特组进行检测,每个比特组长度为 66,如果检测到与预设的块同步头之一相同的比特片段,则指示该比特片段所在位置对应的计数器增加计数值;初始化模块,用于在检测模块进行检测前对 66 个计数器进行初始化;块同步判决器,用于在检测模块完成检测后判断 66 个计数器中是否只有一个计数器的计数值大于 $B_threshold$,如果是,则将该计数器所对应的位置确定为块同步头的位置。

[0058] 同步头位置的计算方法如下:当有且仅有一个计数器 $x(x = 1 \sim N)$ 的数值大于 $B_threshold$ 时,输入数据的第 $x+n \times 66(n = 0 \sim N-1)$ 个比特便是 64b/66b 线路编码块的同步头的开始,此时块同步判决器宣布同步操作完成,同时块同步判决器启动置零操作,将 66 个计数器置零,便于下一次的同步操作。

[0059] 如果块同步判决器判定 66 个计数器的计数值都小于或等于 $B_threshold$,或者 66 个计数器中有至少两个计数器的计数值大于 $B_threshold$,则指示初始化模块对 66 个计数

器重新进行初始化,再指示检测模块对新的 N 个比特组进行检测。

[0060] 另外,检测模块还包括:移位寄存器和比较器。

[0061] 其中,移位寄存器,用于保存当前被检测的两个比特;比较器,用于比较移位寄存器中的两个比特,如果两个比特不同,则指示两个比特所组成的比特片段所对应的计数器增加计数值。

[0062] 在该块同步装置进行工作前,先将 66 个循环计数器均置为 0,接着将接收到的数据的第 1 个比特和第 2 个比特分别送入到一个 2 比特的移位寄存器的地址 1 和地址 2,接着寄存器将所存的数据送入到比较器,如果这 2 个比特的取值不同,则此比较器触发第 1 个计数器使之加 1,如果这 2 个比特取值不同的话,不进行触发操作。

[0063] 然后,移位寄存器将地址 2 中的第 2 个比特数据移位至地址 1,再将接收到的第 3 个比特的数据送入到移位寄存器的地址 2。同样比较器对移位寄存器中的 2 个比特的数据进行比较,如果取值不同的话将触发第 2 个计数器使之加 1,否则,不进行触发计数操作。

[0064] 依次类推,当移位寄存器地址 1 存放第 66 个比特接收数据,地址 2 存放第 67 个比特接收数据时,比较器将这两个比特进行比较后,决定是否对第 66 个计数器进行触发计数操作。接下来,当移位寄存器地址 1 存放第 67 个比特接收数据,地址 2 存放第 68 个比特接收数据时,比较器将这两个比特进行比较后,循环到第 1 个计数器,如果这两个比特数据的值不同,则比较器触发第 1 个计数器使之加 1,否则,不进行触发计数操作。

[0065] 本发明第三实施方式涉及一种块同步方法,与本发明第一实施方式类似,对 N 个比特组中的 L 个比特的数据依次进行检测和分析,区别在于,在本实施方式的检测过程中,在每次比较了第 a 个比特和第 a+1 个比特的取值并对应的将计数器 Bff_cnt[a%L] (其中,令 a=L 时, Bff_cnt[0] 指向第 L 个计数器) 的值加 1 或保持数值的操作后,判断如果有至少两个计数器的计数值大于 B_threshold,则立即停止当前的检测,重新进行块同步,这样可以在检测完 N 个比特组之前尽早结束不可能成功的检测,节省检测时间;另外,第一实施方式中所接收的数据流中各个块的块同步头均为预设的块同步头之一,而在本实施方式中,所接收的数据流中包含块同步头与预设的块同步头不同的块,从而可以使本发明可以适用在有多种块共存的帧中成功找到块同步头的位置。相应地, B_threshold 的计算方法也不同。

[0066] 具体地说,在本实施方式中, $B_threshold = N - B_offset - P$ 。其中, B_offset 同样可以根据块预设的误码率确定,预设的误码率越高, B_offset 值越大,可以灵活设置的 B_offset 可以适应不同的输入误码率; N 同样为所测试的比特组的数目;而 N-P 为块同步头为预设的块同步头的块的数目; P 则为块同步头不属于预设的块同步头的块的数目。根据 B_threshold 判断 L 个计数器中是否有至少两个计数器的计数值大于 B_threshold,如果是,则重新进行块同步。

[0067] FEC 信息帧如图 5 所示,每一帧包含 30 个块,分别为 27 个线路编码块和 3 个校验信息块,每 27 个连续的线路编码块后面续接着 3 个连续的校验块,每一帧相互续接。另外,线路编码块的长度和校验信息块的长度均为 $L = 66$ 比特,其中包含 2 比特的同步头,具体为:线路编码块的同步头取值相反,为“01”或“10”;而校验块的同步头取值相同,为“00”或“11”。

[0068] 根据 $B_threshold = N - B_offset - P$ 计算公式, P 为块同步头不属于预设的块同步

头的块的数目,而如图 5 所示的信息帧中,3 个校验信息块的同步头“00”和“11”不属于预设的块同步头,也就是说, $P = 3$ 。因此,根据设置的 B_offset 值,和已知的 $N = 30$,很容易得到 $B_threshold$ 的值。

[0069] 本实施方式的具体流程如下:

[0070] 步骤 401 至步骤 403 分别与步骤 201 至步骤 203 相类似,在此不再赘述。

[0071] 在步骤 404 中,判断 L 个计数器中是否有至少 2 个计数器的计数值大于 $B_threshold$,如果是,则立即停止当前的检测,并转入返回步骤 401,重新对后续数据进行块同步检测;否则,表示 L 个计数器中有 1 个或 0 个计数器的计数值大于 $B_threshold$,进入步骤 405。

[0072] 步骤 405 至步骤 410 也分别与步骤 204 至步骤 209 相类似,在此不作赘述。

[0073] 本实施方式的块同步方法的流程图还可以采用如图 6 所示的流程图来表示,均不偏离本发明的精神。在进行块同步之前,将表示同步状态的变量 $Block_lock$ 置为“false”,表示开始进行块同步。同时重置计数器,将 66 个计数器 $Bff_cnt[66]$ 全部置零,同时将块计数器 $Block_cnt$ 置零。然后进行块同步测试,执行 $if(input[a] \neq input[a+1])$ 语句,即将输入数据的上一个比特和当前比特进行比较。如果此语句值为真,为合法变换,计数器 $Bff_cnt[a\%66]$ 值加 1, a 的值加 1 (即 $a++$),同时判决 $a\%66$ 是否为 0,如果为 0,块计数器 $Block_cnt$ 加 1。如果此语句为假,为不合法变换, a 的值加 1,计数器 $Bff_cnt[66]$ 不做动作,只有块计数器根据相应条件进行计数。假设测试块同步所需的信息块的块数为 N ,达到块同步所需的门限值为 $threshold$ 。那么测试到一定长度的序列后,根据测试的情况,块同步会出现几种不同的状态:

[0074] (1) 如果块计数器的值小于 N ,并且 66 个计数器中只有一个计数器大于设定门限 $threshold$,那么继续进行测试。

[0075] (2) 如果块计数器的值为 N ,66 个计数器中所有计数器均小于或等于门限 $threshold$,或者多个计数器大于门限 $threshold$,那么宣布块同步操作失败,重新进行同步。

[0076] (3) 如果块计数器的值小于 N ,并且 66 个计数器中所有的计数器都小于或等于门限 $threshold$,那么将继续进行测试。

[0077] (4) 如果块计数器的值小于 N ,并且 66 个计数器中的多个计数器大于门限 $threshold$,宣布块同步操作失败,执行滑动测试。即本次块同步测试立即结束,从下一个测试块开始启动下一次块同步测试。

[0078] (5) 如果块计数器的值为 N ,并且 66 个计数器中当且仅有一个计数器的值大于等于门限值 $threshold$,则宣布同步测试成功,将同步状态变量 $Block_lock$ 置为“true”。

[0079] 通过对类似于如图 5 所示的信息帧的数据流的块同步实验测试,在信道编码的 $BER = 10^{-3}$, $N = 60$, $B_offset = 4$ 时,采用本实施方式可以使得该信息帧正确同步的概率接近 100%,而采用现有技术方案的正确同步的概率为 89.8%;在信道编码的 $BER = 10^{-2}$, N 为 60, $B_offset = 4$ 时,采用本实施方式可以使得该信息帧正确同步的概率为 99.6%,而采用现有技术方案的正确同步的概率为 38.3%。再次说明了本发明的实施方式在信道编码的 BER 比较高的情况下,仍然能够保持高概率的块同步,从而说明了本发明的优越性。

[0080] 本发明第四实施方式涉及一种块同步装置,该装置与第二实施方式所描述的装置

类似,同样包含 L 个计数器、检测模块、初始化模块和块同步判决器。其中,检测模块还包括移位寄存器和比较器。区别在于,在本实施方式中,该装置还包含检测中止模块,如图 7 所示。

[0081] 该模块用于在检测模块对 N 个比特组的检测过程中,判断 L 个计数器中是否有至少两个计数器的计数值大于 B_threshold,如果是,则指示初始化模块对 L 个计数器重新进行初始化,再指示检测模块对新的 N 个比特组进行检测。

[0082] 本发明第五实施方式涉及一种帧同步方法,在完成块同步,确定接收的数据流中线路编码块和校验信息块的起始和终止位置后,进行信息帧的同步。例如,对如图 5 所示的信息帧采用第三实施方式进行块同步成功后,对该数据流的帧进行帧同步如图 8 所示。

[0083] 每一帧包含 $M = 30 (M > 1)$ 个块,分别为 27 个线路编码块和 3 个校验信息块,每 27 个连续的线路编码块后面续接着 3 个连续的校验块,每一帧相互续接。线路编码块的长度和校验信息块的长度均为 $L = 66$ 比特,其中包含 2 比特的同步头,具体为:线路编码块的同步头取值相反,为“01”或“10”;而校验块的同步头取值相同,为“00”或“11”。帧同步方法如图 9 所示,其具体流程如下。

[0084] 在步骤 901 中,对 K 个比特组进行检测,每个比特组包括 $M = 30$ 个块, $K > 1$,其中,对每个比特组进行检测时,读取该比特组内 M 个块的块同步头组成接收序列。

[0085] 例如如图 10 所示,用 h11 表示接收的第 1 个数据块的第 1 个同步头, h12 表示接收的第 1 个数据块的第 2 个同步头,相应的 h301 和 h302 分别表示接收的第 30 个数据块的第 1 和第 2 个同步头。从某一个块开始,连续对 FEC 信息帧的 $M = 30$ 个块进行同步测试,这 30 个数据块中肯定包含 27 个线路编码块和 3 个校验块。将 h11 和 h12 送入到一个循环移位寄存器的前两个位置,然后,间隔 64 比特,找到 h21 和 h22,并将 h21 和 h22 送入循环移位寄存器的随后的两个位置。依次类推,直到将 h11 至 h302 依次存到循环移位寄存器中。

[0086] 在步骤 902 中,将接收的序列与参考序列进行相关运算,得到循环移动 0 比特的相关值,并记录该相关值。

[0087] 其中,参考序列由线路编码块同步头和校验信息块同步头两部分组成,线路编码块的同步头可以设置为“01”或“10”中的任何一种,校验信息块同步头则可以设置为“00”或“11”中的任何一种。因此,可以将参考序列的线路编码块的同步头部分全部设置成“01”,3 个校验信息块同步头为:00, 11, 11。

[0088] 同步相关运算的基本思想是使用一种运算方法来对接收同步头和参考序列的同步头的相关程度进行衡量。比如参考同步头为“00”,如果接收同步头也为“00”的话,则表明两者完全相关,如果接收同步头为“11”的话,表明两者不相关,或相关性很差,如果接收同步头为“01”或“10”的话,表明两者具有一定的相关性。

[0089] 一种同步相关运算方法如表 1 所示,在该表中,罗列了 16 种可能的相关运算的情况及其对应的结果,通过查询该表即可获得相关值。具体地说,接收的块同步头为“00”,与参考序列对应值为“00”、“11”、“01”、“10”的相关值分别为 2、0、1、1;接收的块同步头为“11”,与参考序列对应值为“00”、“11”、“01”、“10”的相关值分别为 0、2、1、1;接收的块同步头为“01”,与参考序列对应值为“00”、“11”、“01”、“10”的相关值分别为 1、1、2、2;接收的块同步头为“10”,与参考序列对应值为“00”、“11”、“01”、“10”的相关值分别为 1、1、2、2。

[0090] 表 1

[0091]

接收的块同步头	参考序列	相关值
0 0	0 0	2
0 0	1 1	0
0 0	0 1/1 0	1
1 1	0 0	0
1 1	1 1	2
1 1	0 1/1 0	1
0 1/1 0	0 0	1
0 1/1 0	1 1	1
0 1/1 0	0 1/1 0	2

[0092] 以循环移位寄存器中的同步头序列“01 10 00 11 11 01 10 01 10”，参考序列中的同步头序列“01 01 01 01 01 01 00 11 11”为例。

[0093] 根据表 1 所示的同步相关运算法则，寄存器中的第 1 对同步头“01”与参考序列中的第 1 对同步头“01”的相关值为 2，寄存器中第 1 对同步头“10”与参考序列中的第 1 对同步头“01”的相关值为 2，依次进行计算，然后将各个相关值相加得到寄存器中的同步序列和参考序列的相关值为 12。

[0094] 另一种同步相关运算方法为采用公式一进行计算：

$$\gamma = \sum_{i=1}^M (2 - |(c_{i1} - r_{i1}) + (c_{i2} - r_{i2})|) \text{公式 (1)}$$

[0095] 其中， γ 为接收序列和参考序列的相关值， c_{i1} 、 c_{i2} 分别为接收序列第 i 的数据块的第 1、2 个同步头， r_{i1} 、 r_{i2} 分别为参考序列中与接收序列第 i 的数据块相对应位置的数据。同样以循环移位寄存器中的同步头序列“01 10 00 11 11 01 1001 10”，参考序列中的同步头序列“01 01 01 01 01 01 00 11 11”为例进行计算，可以发现与采用表 1 运算方法得到的结果相同。

[0096] 在步骤 903 中，以块同步头的长度 2 比特为单位对接收序列进行所有可能的循环移位，该循环移位可以是向左或向右循环移位 2 比特，例如如图 10 所示向右循环移位 2 为后，第 30 的数据块的同步头 h301 和 h302 移到寄存器的前 2 个位置。

[0097] 在步骤 904 中，判断循环移位是否结束，如果经过步骤 903 移位后的序列与移位 0 比特的序列相同了，也即第 M 次移位后，则判断循环移位结束，进入步骤 905；如果循环移位次数小于 M ，循环移位没有结束，则转入步骤 902，对移位后的序列同参考序列进行相关运

算,得到对应的相关值并保存该值。

[0098] 当然,也可以对移位次数进行计数,对第 $M-1$ 次移位后的序列同参考序列进行相关运算并保存所得到的相关值后,就可以结束对该比特组的移位操作,同样转入步骤 905。

[0099] 在步骤 905 中,判断是否已经将 K 个比特组都分别进行了读取块同步头,对序列的移位和相关运算的操作,得到了 K 个比特组的序列对应各循环位置的相关值,如果是,则进入步骤 907;否则,进入步骤 906,继续读取 K 个比特组中的下一个比特组的 M 个同步头组成序列,并进入步骤 902,将该序列同参考序列进行相关运算,如此循环往复,直到将 K 个比特组对应的 K 个序列,在每个循环位置上与参考序列的 M 个相关值都得到为止。

[0100] 在步骤 907 中,得到 K 个比特组对应的 K 个序列,在每个循环位置上与参考序列的相关值后,将循环位置相同的 K 个相关值相加,得到 M 个相关和。

[0101] 例如,检测 3 个比特组(即 $K = 3$),每个比特组中包含的块数 $M = 6$,第 1、2、3 个帧对应的循环移位 6 比特与参考序列的相关值分别为 54、48、32,则该信息帧对应循环移位 6 比特的相关和为 $54+48+32 = 134$ 。

[0102] 在步骤 908 中,判断所得到的 M 个相关和中,是否只有一个相关和大于帧门限值 ($F\text{-threshold}$), $F\text{-threshold} = 2M \times K - \text{帧偏置值}(F\text{-offset})$ 。其中, $F\text{-offset}$ 值根据帧中预设的误码率确定,预设的误码率越高,该 $F\text{-offset}$ 越大,通过灵活设置 $F\text{-offset}$ 可以适应不同的输入误码率。

[0103] 如果只有一个相关和大于 $F\text{-threshold}$,则进入步骤 909,根据该相关和所对应的循环移位次数得到该帧头的位置,完成帧同步。例如,如果只有向右循环移位 y 比特对应的相关和大于 $F\text{-threshold}$,则可以确定 K 个比特组中的第 $M-y/2$ 个数据块为信息帧的起始位置;如果只有向左循环移位 y 比特对应的相关和大于 $F\text{-threshold}$,则可以确定 K 个比特组中的第 $y/2$ 个数据块为信息帧的起始位置。

[0104] 如果所有的相关和都小于或等于 $F\text{-threshold}$,或者有至少两个相关和大于 $F\text{-threshold}$,则返回步骤 901,重新进行帧同步。

[0105] 对由 M 个块同步头所组成的接收序列进行 $M-1$ 次循环位移,可以得到接收序列各种可能的循环位移结果,将这些循环位移结果与参考序列进行相关运算,可以用相关值定量地指示各种循环位移结果与参考序列接近的程度,通过将 K 个比特组中对应相同循环移位次数的相关值相加,在输入误码率比较高时,仍然能够可靠地知道哪一种循环位移结果与参考序列最为接近,从而准确知道帧头的位置。

[0106] 本发明第六实施方式涉及一种帧同步装置,如图 11 所示,包含检测单元、累加器、 M 个寄存器和帧同步判断器。

[0107] 其中,检测单元,用于对 K 个比特组进行检测,每个比特组包括 M 个块,其中,对每个比特组进行检测时,读取该比特组内 M 个块的块同步头组成接收序列,以块同步头的长度为单位对接收序列进行所有可能的循环移位,将该接收序列经循环移位所有可能得到的排列与参考序列进行相关运算,记录每次相关运算所得的相关值, $K > 1, M > 1$;

[0108] 累加器,用于将检测单元得到的 K 个比特组中对应相同循环移位次数的相关值相加,得到 M 个相关和;

[0109] M 个寄存器,分别用于存储累加器得到的 M 个相关和;

[0110] 帧同步判断器,用于判断 M 个寄存器中是否只有一个相关和大于 $F\text{-threshold}$,其

中, $F\text{-threshold} = 2M \times K - F\text{-offset}$, 如果是, 则根据该相关和所对应的循环移次数得到帧头的位置。

[0111] 如果帧同步判断器判定 M 个寄存器中所有的相关和都小于或等于 $F\text{-threshold}$, 或者有至少两个相关和大于 $F\text{-threshold}$, 则指示检测单元对新的 K 个比特组进行检测。

[0112] 另外, 检测单元还包括循环移位寄存器、参考序列存储器和相关运算器。

[0113] 具体地说, 循环移位寄存器, 由 $2M$ 个比特组成, 用于保存一个比特组内 M 个块的块同步头所组成的接收序列, 该循环移位寄存器每次循环移位两个比特; 参考序列存储器, 用于保存由 $2M$ 个比特组成的参考序列; 相关运算器, 用于对循环移位寄存器和参考序列存储器进行按位相关运算, 得到相关值。

[0114] 用相关值定量地指示各种循环移位结果与参考序列接近的程度, 通过将 K 个比特组中对应相同循环移次数的相关值相加, 在输入误码率比较高时, 仍然能够可靠地知道哪一种循环移位结果与参考序列最为接近, 从而准确知道帧头的位置。

[0115] 通过实验测试, 对于结构如图 5 所示的信息帧, 当 $K = 3, F\text{-offset} = 5, BER = 10^{-3}$ 时, 采用本实施方式对 3 个信息帧的数据进行帧同步的正确帧同步的概率接近 100%, 而采用现有技术正确帧同步的概率为 78.7%; 当 $K = 3, F\text{-offset} = 5, BER = 10^{-2}$ 时, 采用本实施方式正确帧同步的概率为 96.7%, 而采用现有技术正确帧同步的概率仅为 9.1%。从测试结果可以看出本发明的设计方案与现有技术相比, 在 BER 比较高时, 能够维持较高的正确同步率。

[0116] 综上所述, 在本发明的实施方式中, 检测 N 个比特组, 对每一个比特组中各个位置出现与块同步头相同的比特片段的次数进行统计, 如果只有一个位置的统计值大于第一门限, 则该位置即为块同步头的位置。在输入误码率比较高时只是使各个位置的统计值较为接近, 但块同步头位置的统计值明显大于其它位置的统计值的概率依然很大, 因此仍然能够准确地完成同步。

[0117] 在检测过程中, 如果有至少两个计数器的计数值大于第一门限, 则重新进行块同步, 这样可以在检测完 N 个比特组之前尽早结束不可能成功的检测, 节省检测时间。

[0118] 第一门限可以设置成 N - 块偏置值, 通过灵活设置块偏置值可以适应不同的输入误码率。

[0119] 第一门限也可以设置成 N - 块偏置值 $-P$, 在 N 个块中有 P 个块的块同步头不属于预设的块同步头之一, 从而可以在有多种块共存的帧中, 成功找到块同步头的位置。

[0120] 通过比较两个相邻比特是否互异, 可以在块同步头只有两个比特, 且预设的块同步头为“10”或“01”时, 快速地判断两个相邻比特是否为预设的块同步头之一。

[0121] 对由 M 个块同步头所组成的接收序列进行 $M-1$ 次循环移位, 可以得到接收序列各种可能的循环移位结果, 将这些循环移位结果与参考序列进行相关运算, 可以用相关值定量地指示各种循环移位结果与参考序列接近的程度, 通过将 K 个比特组中对应相同循环移次数的相关值相加, 在输入误码率比较高时, 仍然能够可靠地知道哪一种循环移位结果与参考序列最为接近, 从而准确知道帧头的位置。

[0122] 虽然通过参照本发明的某些优选实施方式, 已经对本发明进行了图示和描述, 但本领域的普通技术人员应该明白, 可以在形式上和细节上对其作各种改变, 而不偏离本发明的精神和范围。

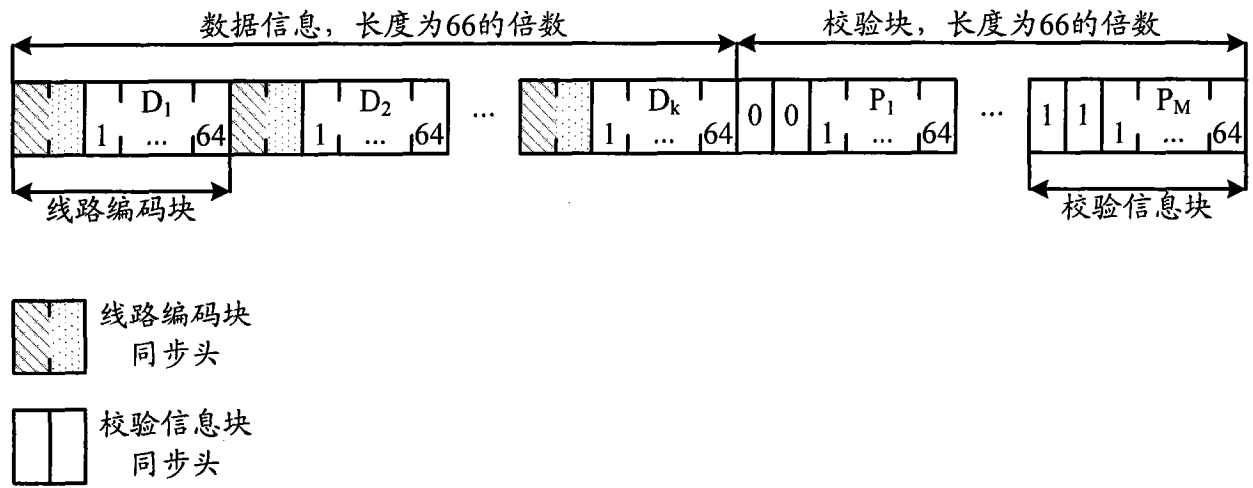


图 1

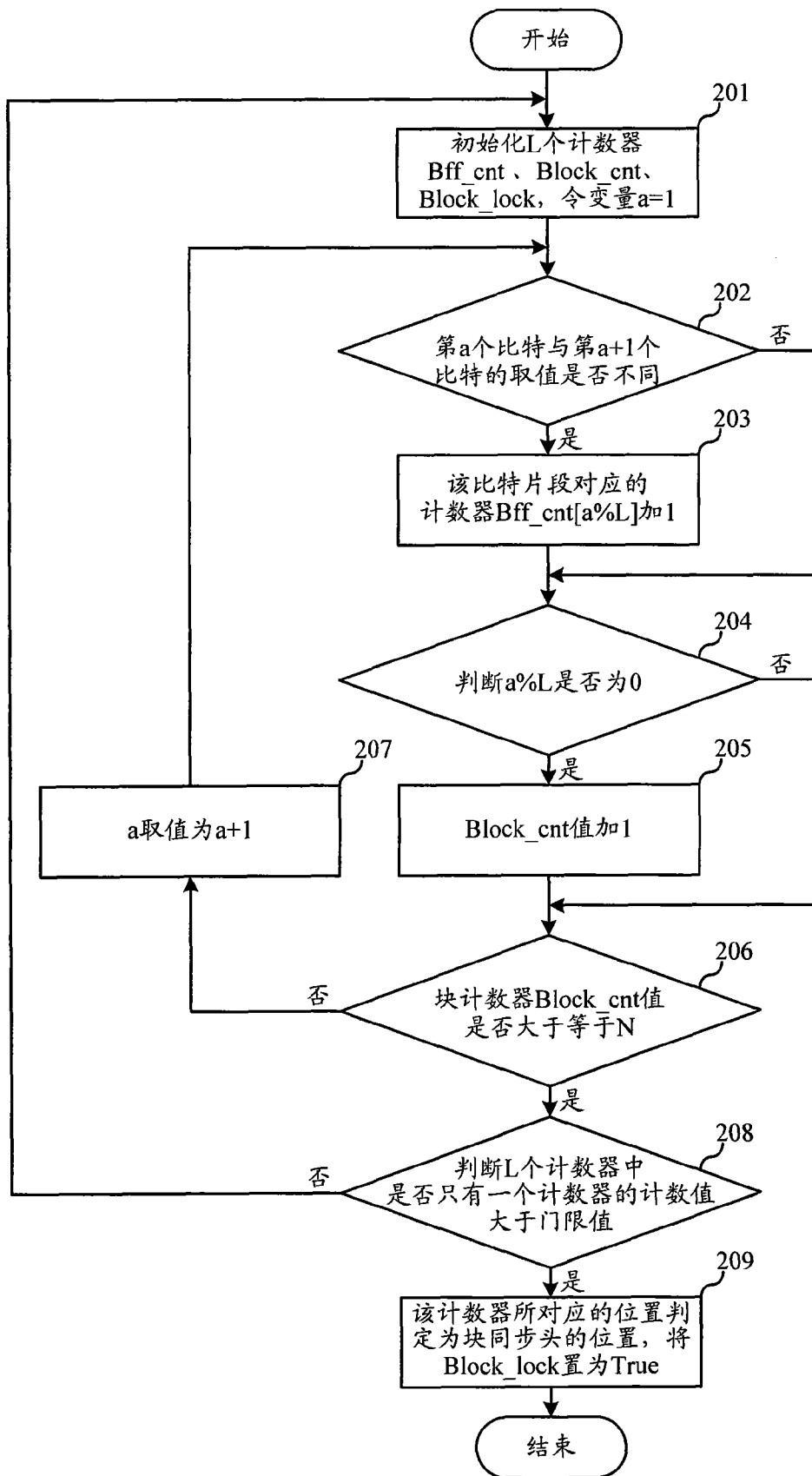


图 2

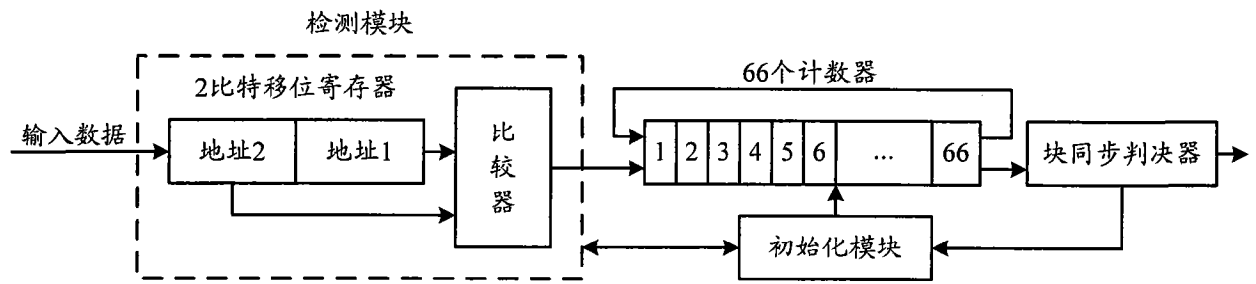


图 3

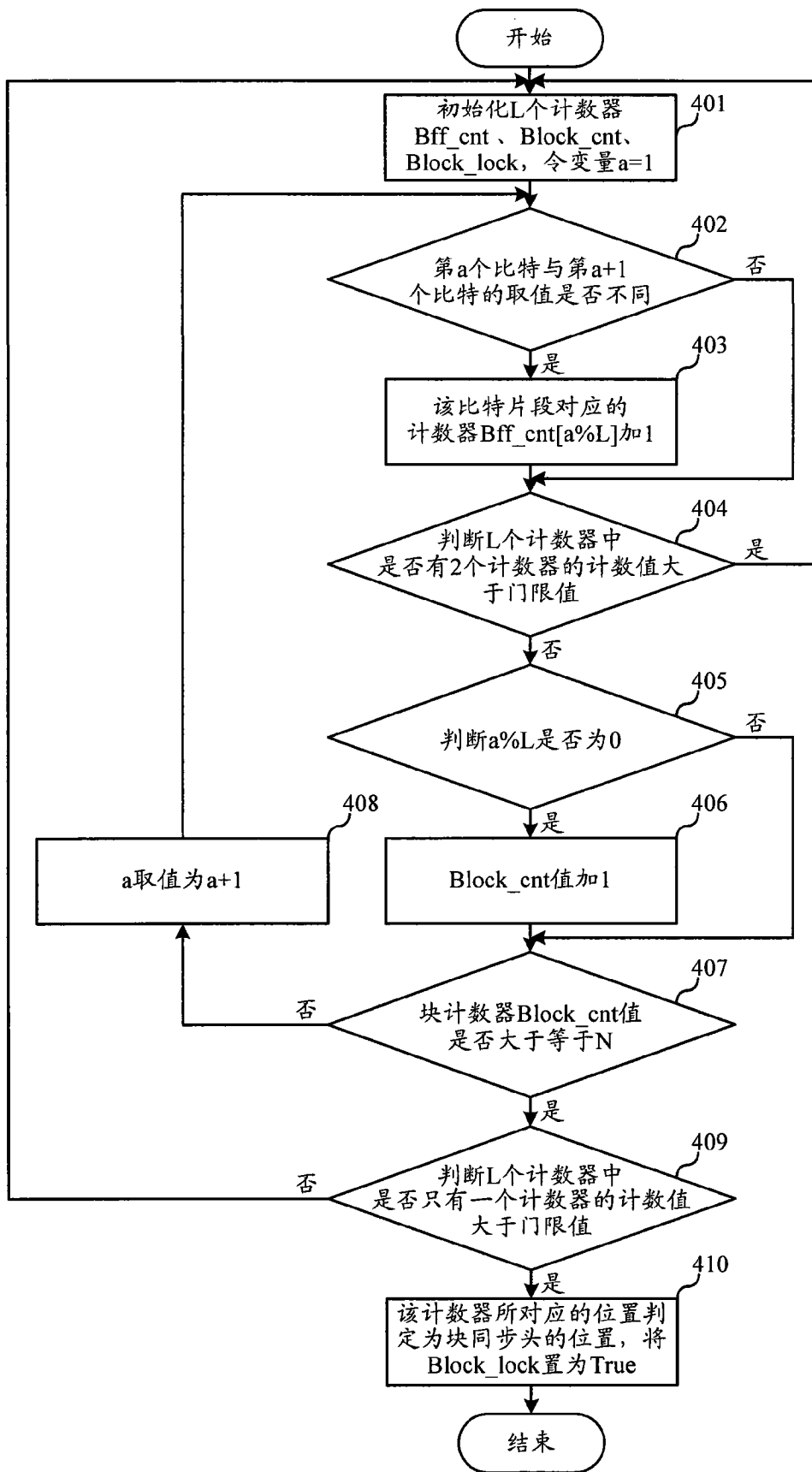


图 4

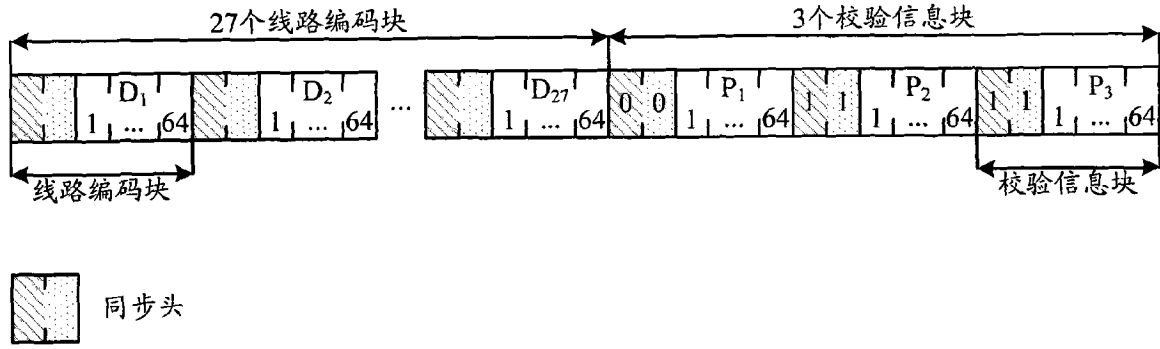


图 5

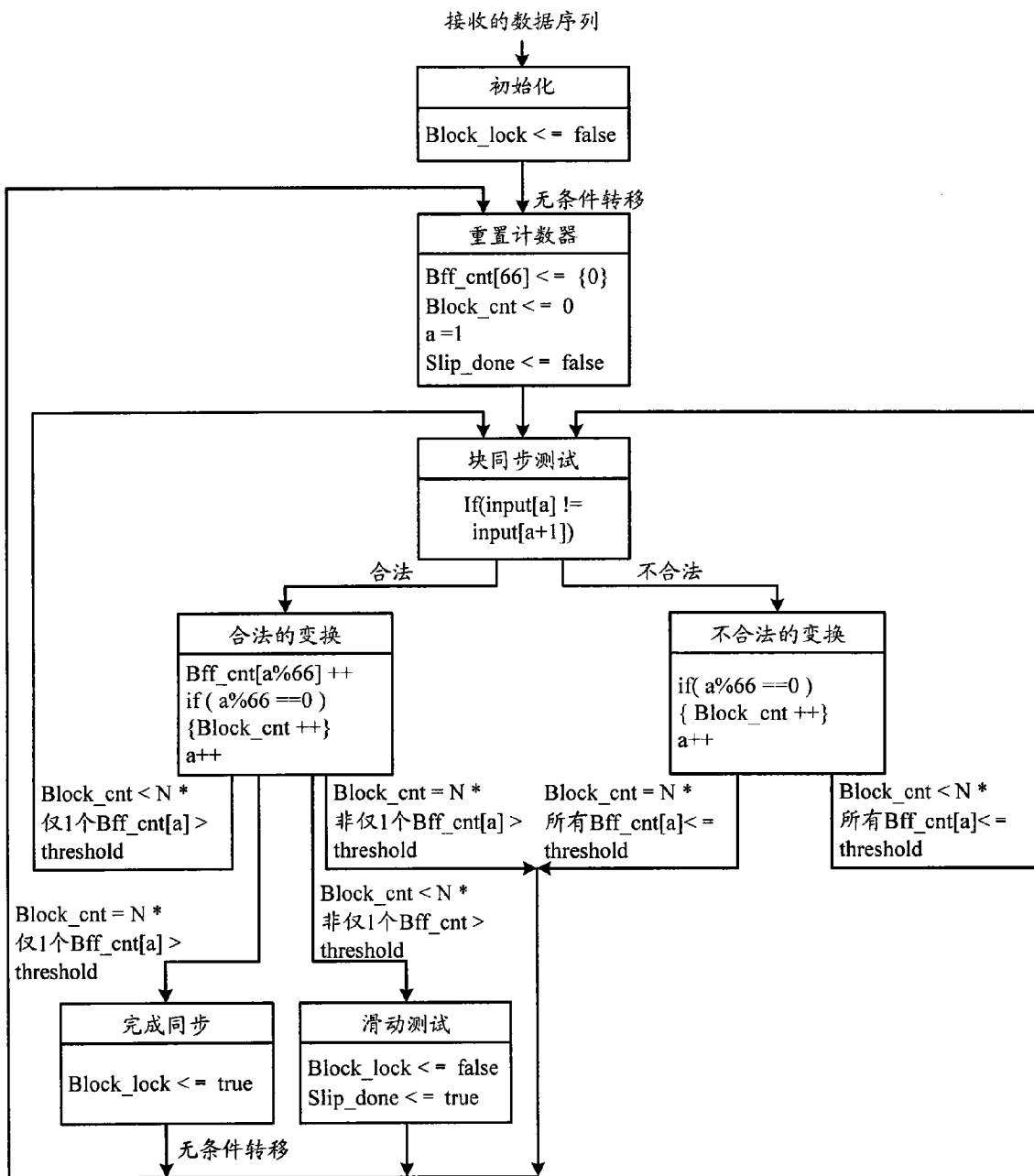


图 6

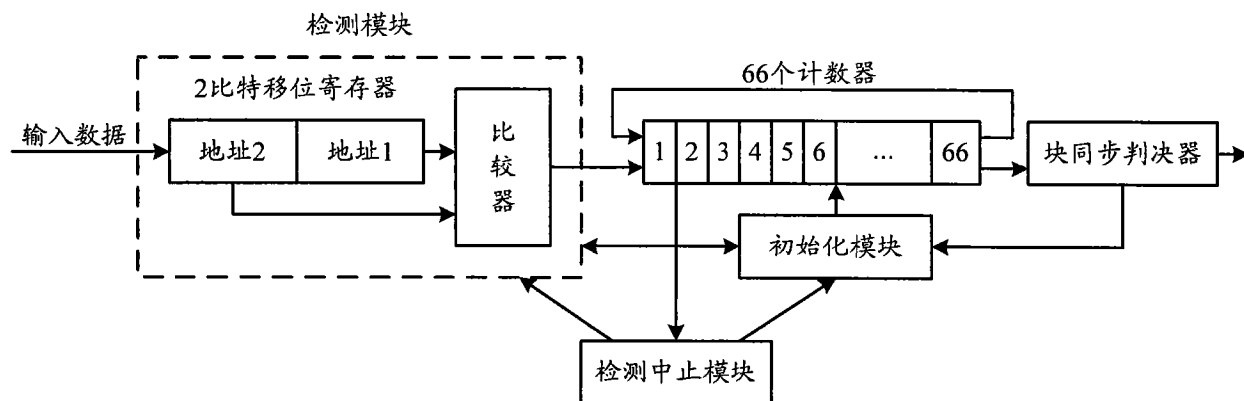


图 7

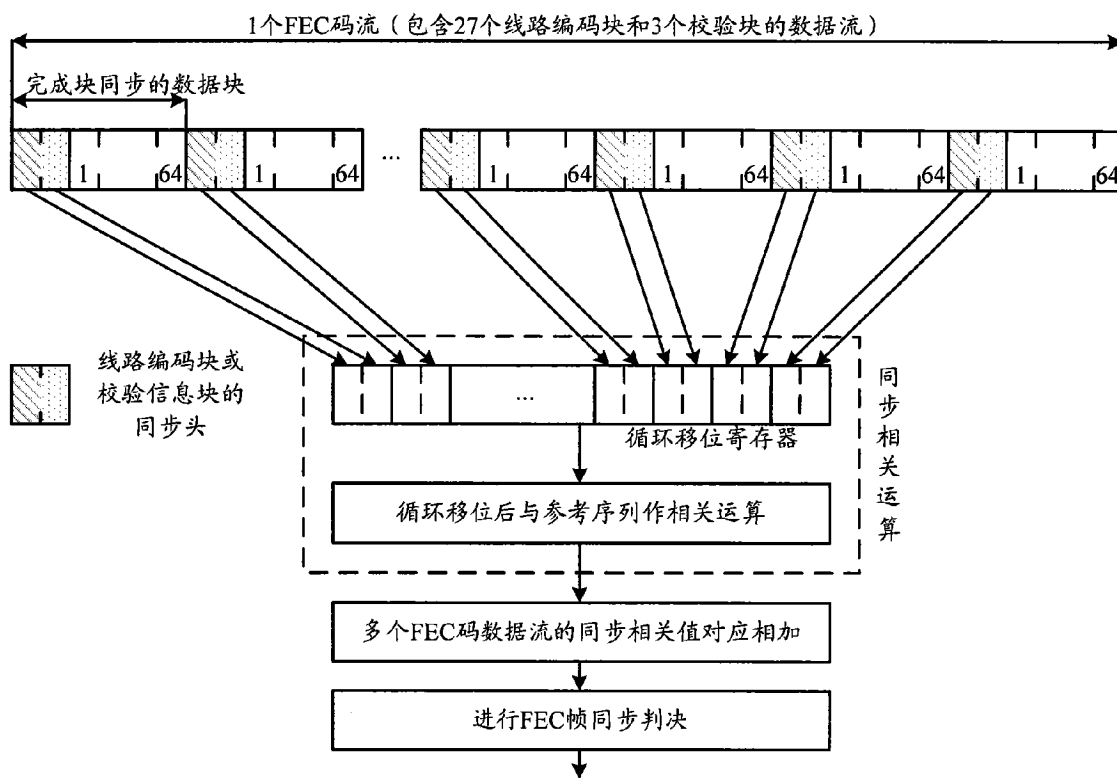


图 8

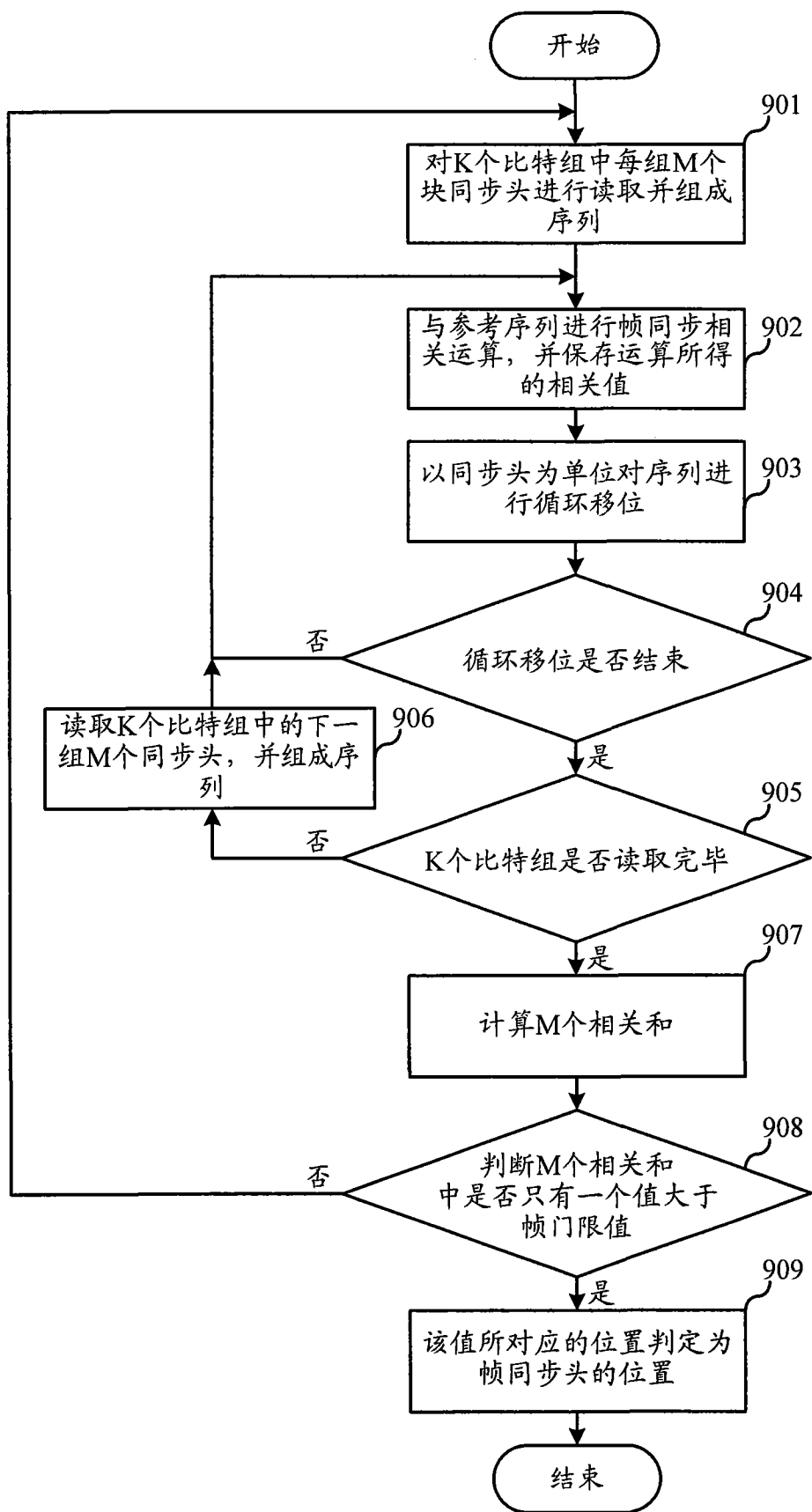


图 9

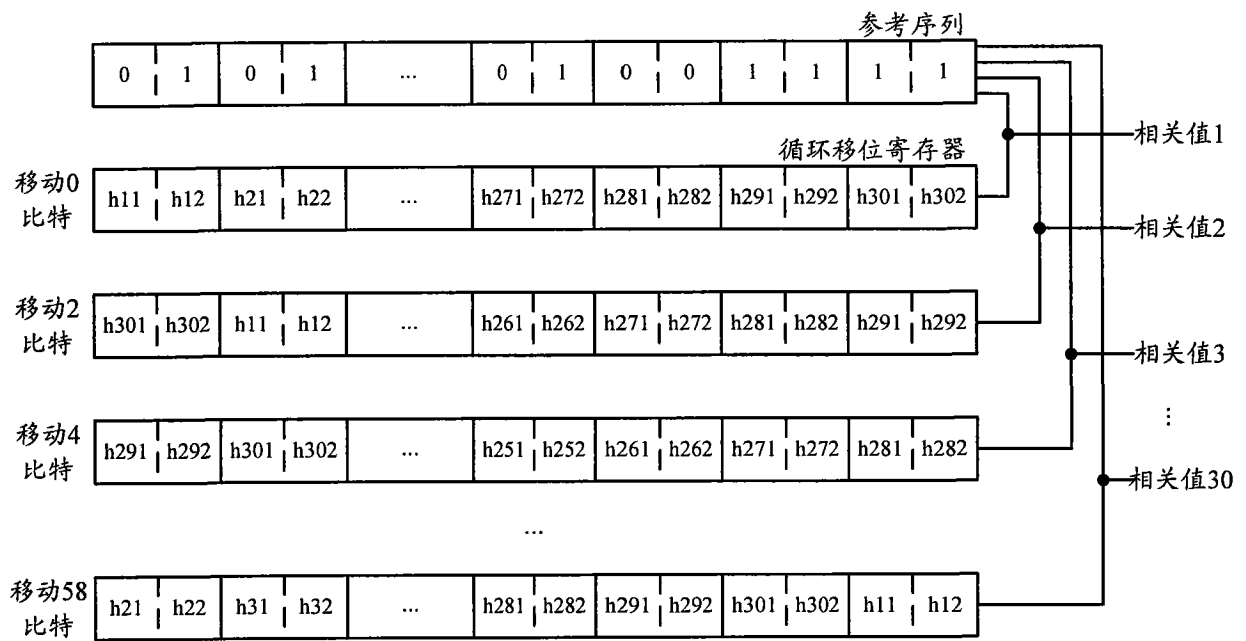


图 10

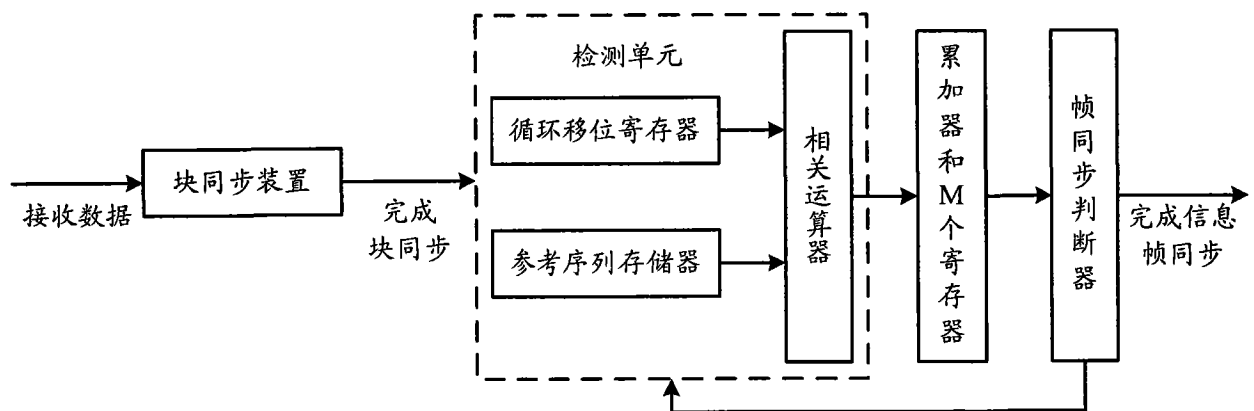


图 11