



(12) 发明专利申请

(10) 申请公布号 CN 102468902 A

(43) 申请公布日 2012. 05. 23

(21) 申请号 201010532617. X

(22) 申请日 2010. 11. 01

(71) 申请人 普天信息技术研究院有限公司

地址 100080 北京市海淀区海淀北二街 6 号

(72) 发明人 周晟 刘富芝

(74) 专利代理机构 北京德琦知识产权代理有限公司

公司 11018

代理人 王一斌 王琦

(51) Int. Cl.

H04L 1/00 (2006. 01)

H04L 1/06 (2006. 01)

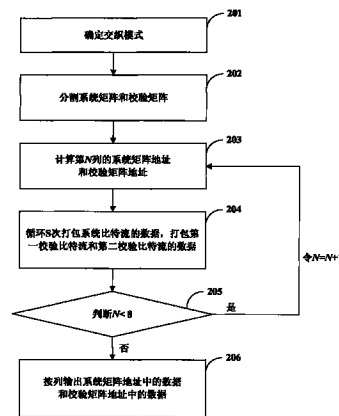
权利要求书 3 页 说明书 10 页 附图 3 页

(54) 发明名称

LTE 系统 Turbo 编码速率匹配 / 解速率匹配的方法

(57) 摘要

一种 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 该方法包括: 根据编码块的数据长度 K 确定交织模式; 由系统比特流得到系统矩阵, 第一校验比特流 P1 与第二校验比特流 P2 交替存放得到校验矩阵, 从系统矩阵的第 1 列开始每 8 列为一个子系统矩阵, 从校验矩阵的第 1 列开始每 8 列为一个子校验矩阵; 确定系统矩阵地址和校验矩阵地址; 系统比特流的数据打包后放置于地址中, 校验比特流数据打包后置于校验矩阵的地址中; 按列输出系统矩阵地址中和校验矩阵地址中的数据, 得到速率匹配 / 解速率匹配后的比特流。应用本发明实施例以后, 能够加快速率匹配 / 解速率匹配的速度。



1. 一种 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 该方法包括:

根据编码块的数据长度 K 确定交织模式;

由系统比特流得到系统矩阵, 第一校验比特流 $P1$ 与第二校验比特流 $P2$ 交替存放得到校验矩阵, 从系统矩阵的第 1 列开始每 8 列为一个子系统矩阵, 从校验矩阵的第 1 列开始每 8 列为一个子校验矩阵;

从 $N = 1$ 开始并按 1 递增, 直至 $N = 8$, 依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址, 从 $N = 1$ 开始并按 1 递增, 直至 $N = 8$, 依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址;

每次提取中, 选择 4 个系统比特流的字打包后按照预定规则置于所述系统矩阵地址中, 选择 4 个 $P1$ 的字和 4 个 $P2$ 的字打包后按照预定规则置于所述校验矩阵的地址中, 该选择包括 S 次循环, S 等于子系统矩阵的行数 R 减 1 后除以 4 向下取整, R 等于 K 加 4 后除以 32 向上取整;

按列输出系统矩阵地址中和校验矩阵地址中的数据, 得到速率匹配 / 解速率匹配后的比特流。

2. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 所述根据编码块的数据长度 K 确定交织模式包括, K 对 32 取余数, 由所述余数确定交织模式。

3. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 所述依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址包括, 根据交织模式确定交织索引, 由交织索引依次偏移每个子系统矩阵的第 N 列中每个数据的地址得到该列每个数据的中间偏移地址, 再根据交织模式和 N 整体偏移所述中间偏移地址得到系统矩阵地址;

所述依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址包括, 根据交织模式确定交织索引, 由交织索引依次偏移每个子校验矩阵的第 N 列中每个数据的地址得到该列每个数据的中间偏移地址, 再根据交织模式和 N 整体偏移所述中间偏移地址得到校验矩阵地址。

4. 根据权利要求 3 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 所述由交织索引依次偏移每个子系统矩阵的第 N 列中每个数据的地址得到中间偏移地址之前进一步包括, 计算系统矩阵的起始列位置, 系统矩阵的起始位置等于 k_0 ,

$$k_0 = R \cdot \left(2 \cdot \left\lceil \frac{N_{cb}}{8R} \right\rceil \cdot RV + 2 \right), N_{cb} \text{ 为速率匹配软 Buffer 大小, } RV \text{ 是冗余版本参数。}$$

5. 根据权利要求 4 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 当 k_0 大于 32, 系统矩阵的起始列位置等于 $\lfloor k'_0 \rfloor + 16$ 。

6. 根据权利要求 3 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法, 其特征在于, 所述根据交织模式和 N 整体偏移所述中间偏移地址得到系统矩阵地址包括, 根据交织模式和 N 确定整体偏移量, 然后根据整体偏移量整体偏移所述中间偏移地址得到系统矩阵地址;

所述根据交织模式和 N 整体偏移所述中间偏移地址得到校验矩阵地址包括,根据交织模式和 N 确定整体偏移量,然后根据整体偏移量整体偏移所述中间偏移地址得到校验矩阵地址。

7. 根据权利要求 6 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征在于,所述根据交织模式和 N 确定整体偏移量包括,由交织模式确定填充比特,整体偏移量 H 等于 32 减去填充比特后与 P1 的第 N 个数据的和。

8. 根据权利要求 6 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征在于,所述根据交织模式和 N 确定整体偏移量包括,由交织模式确定多余比特,整体偏移量 H 等于 P1 的第 N 个数据与填充比特的差。

9. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征在于,所述选择 4 个系统比特流的字打包后按照预定规则置于所述系统矩阵地址包括,

从第 0 个系统字开始,每隔 8 个字取出比特流的字,得到第一系统字、第二系统字、第三系统字和第四系统字;分别取所述四个系统字的最高数据组成第一系统输出字,次高数据组成第二系统输出字,次低数据组成第三系统输出字,最低数据组成第四系统输出字;

将所述第一系统输出字放置于所述系统矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将所述第二系统输出字放置于所述系统矩阵第 S 行第 9 个数据至第 12 个数据的地址中,将所述第三系统输出字放置于所述系统矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将所述第四系统输出字放置于所述系统矩阵第 S 行第 13 个数据至第 16 个数据的地址中。

10. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征在于,所述选择 4 个 P1 的字和 4 个 P2 的字打包后按照预定规则置于所述校验矩阵的地址中包括,

顺序从 P1 取出第 0 个字和第 8 个字,依次记为第一校验字和第二校验字,顺序从 P2 取出第 1 个字和第 9 个字,依次记为第三校验字和第四校验字;

分别取第一校验字至第四校验字的最高数据组成第一校验输出字,次高数据组成第二校验输出字,次低数据组成第三校验输出字,最低数据组成第四校验输出字;

将所述第一校验输出字放置于所述校验矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将所述第二校验输出字放置于所述校验矩阵第 S 行第 9 个数据至第 12 个数据的地址中,将所述第三校验输出字放置于所述校验矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将所述第四校验输出字放置于所述校验矩阵第 S 行第 13 个数据至第 16 个数据的地址中;

然后,再顺序从 P1 取出第 16 个字和第 24 个字,依次记为第五校验字和第六校验字,顺序从 P2 取出第 17 个数据和第 25 个数据,依次记为第七校验字和第八校验字;

分别取第五校验字至第八校验字的最高数据组成第五校验输出字,次高数据组成第六校验输出字,次低数据组成第七校验输出字,最低数据组成第八校验输出字;

将所述第五校验输出字放置于所述校验矩阵第 S+1 行第 1 个数据至第 4 个数据的地址中,将所述第六校验输出字放置于所述校验矩阵第 S+1 行第 9 个数据至第 12 个数据的地址中,将所述第七校验输出字放置于所述校验矩阵第 S+1 行第 5 个数据至第 8 个数据的地址中,将所述第八校验输出字放置于所述校验矩阵第 S+1 行第 13 个数据至第 16 个数据的地址中。

11. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征

在于,当 S 次循环后存在剩余数据时,每次取一个剩余数据放置于所述剩余数据的地址中。

12. 根据权利要求 1 所述 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,其特征在于,当 N 等于 8 进一步包括,根据交织模式填充系统数据的多余比特,根据交织模式填充校验数据的多余比特。

LTE 系统 Turbo 编码速率匹配 / 解速率匹配的方法

技术领域

[0001] 本发明涉及通信技术领域,更具体地,涉及 LTE 系统 Turbo 编码速率匹配 / 解速率匹配的方法。

背景技术

[0002] 长期演进 (LTE) 是 3G 通信技术的长期演进,为未来的无线通信系统提供了更高的传输速率,其高速的码率给基站和终端的基带处理带来了沉重的负担。对于 LTE 技术中的基带处理而言,如何加快比特级的数据处理速度,尤其是传输信道的速率匹配处理速度是整个基带处理的瓶颈之一。

[0003] 现有 Turbo 编码的传输信道速率匹配过程如附图 1 所示。发送端原始的比特流经过 Turbo 编码后得到系统比特流 $d_k^{(0)}$ 、第一校验比特流 $d_k^{(1)}$ 、第二校验比特流 $d_k^{(2)}$ 共计三路数据。 $d_k^{(0)}$ 、 $d_k^{(1)}$ 和 $d_k^{(2)}$ 三路比特流的长度相同,比特流的长度等于 $K+4$, K 是编码块的数据长度,4 是尾比特。尾比特是经过 Turbo 编码剩余的比特。 $d_k^{(0)}$ 、 $d_k^{(1)}$ 和 $d_k^{(2)}$ 三路比特流分别输入子块交织器,即比特流送入一个 R 行,32 列的矩阵中,逐行写入,再进行列间置换,然后逐行读出分别得到与 $d_k^{(0)}$ 相对应的输出比特流 $v_k^{(0)}$,与 $d_k^{(1)}$ 相对应的输出比特流 $v_k^{(1)}$,与 $d_k^{(2)}$ 相对应的输出比特流 $v_k^{(2)}$ 。 $v_k^{(0)}$ 、 $v_k^{(1)}$ 、 $v_k^{(2)}$ 进入比特收集模块。在比特收集模块中,收集的方式是系统比特流在前,第一校验比特流与第二校验比特流交替存放,构成一个完整的比特流 w_k 。再根据速率匹配的起始位置和速率匹配输出的长度,裁剪或者重复取数,直到满足输出长度要求输出比特流 e_k 至终端。

[0004] 根据上述整体流程分析,不需要等到系统比特流 $d_k^{(0)}$ 、第一校验比特流 $d_k^{(1)}$ 、第二校验比特流 $d_k^{(2)}$ 同时到达才开始速率匹配操作,而是将 Turbo 后编码的各个比特流编码后分别进行速率匹配。由于需要反复读取内存中比特流中的数据,因此上述速率匹配的处理速度较低。

发明内容

[0005] 本发明实施例提出一种 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,能够加快速率匹配 / 解速率匹配的速度。

[0006] 一种 LTE 系统中 Turbo 编码速率匹配 / 解速率匹配的方法,该方法包括:

[0007] 根据编码块的数据长度 K 确定交织模式;

[0008] 由系统比特流得到系统矩阵,第一校验比特流 $P1$ 与第二校验比特流 $P2$ 交替存放得到校验矩阵,从系统矩阵的第 1 列开始每 8 列为一个子系统矩阵,从校验矩阵的第 1 列开始每 8 列为一个子校验矩阵;

[0009] 从 $N = 1$ 开始并按 1 递增,直至 $N = 8$,依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址,从 $N = 1$ 开始并按 1 递增,直至 $N = 8$,依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址;

[0010] 每次提取中,选择 4 个系统比特流的字打包后按照预定规则置于所述系统矩阵地址中,选择 4 个 P1 的字和 4 个 P2 的字打包后按照预定规则置于所述校验矩阵的地址中,该选择包括 S 次循环,S 等于子系统矩阵的行数 R 减 1 后除以 4 向下取整,R 等于 K 加 4 后除以 32 向上取整;

[0011] 按列输出系统矩阵地址中和校验矩阵地址中的数据,得到速率匹配 / 解速率匹配后的比特流。

[0012] 所述根据编码块的数据长度 K 确定交织模式包括,K 对 32 取余数,由所述余数确定交织模式。

[0013] 所述依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址包括,根据交织模式确定交织索引,由交织索引依次偏移每个子系统矩阵的第 N 列中每个数据的地址得到该列每个数据的中间偏移地址,再根据交织模式和 N 整体偏移所述中间偏移地址得到系统矩阵地址;

[0014] 所述依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址包括,根据交织模式确定交织索引,由交织索引依次偏移每个子校验矩阵的第 N 列中每个数据的地址得到该列每个数据的中间偏移地址,再根据交织模式和 N 整体偏移所述中间偏移地址得到校验矩阵地址。

[0015] 所述由交织索引依次偏移每个子系统矩阵的第 N 列中每个数据的地址得到中间偏移地址之前进一步包括,计算系统矩阵的起始列位置,系统矩阵的起始位置等于 k_0 ,

$$k_0 = R \cdot \left(2 \cdot \left\lfloor \frac{N_{cb}}{8R} \right\rfloor \cdot RV + 2 \right), N_{cb} \text{ 为速率匹配软 Buffer 大小, } RV \text{ 是冗余版本参数。}$$

[0016] 当 k_0 大于 32,系统矩阵的起始列位置等于 $k'_0, k'_0 = \frac{k_0}{2} + 16$ 。

[0017] 所述根据交织模式和 N 整体偏移所述中间偏移地址得到系统矩阵地址包括,根据交织模式和 N 确定整体偏移量,然后根据整体偏移量整体偏移所述中间偏移地址得到系统矩阵地址;

[0018] 所述根据交织模式和 N 整体偏移所述中间偏移地址得到校验矩阵地址包括,根据交织模式和 N 确定整体偏移量,然后根据整体偏移量整体偏移所述中间偏移地址得到校验矩阵地址。

[0019] 所述根据交织模式和 N 确定整体偏移量包括,由交织模式确定填充比特,整体偏移量 H 等于 32 减去填充比特后与 P1 的第 N 个数据的和。

[0020] 所述根据交织模式和 N 确定整体偏移量包括,由交织模式确定多余比特,整体偏移量 H 等于 P1 的第 N 个数据与填充比特的差。

[0021] 所述选择 4 个系统比特流的字打包后按照预定规则置于所述系统矩阵地址包括,

[0022] 从第 0 个系统字开始,每隔 8 个字取出比特流的字,得到第一系统字、第二系统字、第三系统字和第四系统字;分别取所述四个系统字的最高数据组成第一系统输出字,次高数据组成第二系统输出字,次低数据组成第三系统输出字,最低数据组成第四系统输出字;

[0023] 将所述第一系统输出字放置于所述系统矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将所述第二系统输出字放置于所述系统矩阵第 S 行第 9 个数据至第 12 个数据的地址

中,将所述第三系统输出字放置于所述系统矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将所述第四系统输出字放置于所述系统矩阵第 S 行第 13 个数据至第 16 个数据的地址中。

[0024] 所述选择 4 个 P1 的字和 4 个 P2 的字打包后按照预定规则置于所述校验矩阵的地址中包括,

[0025] 顺序从 P1 取出第 0 个字和第 8 个字,依次记为第一校验字和第二校验字,顺序从 P2 取出第 1 个字和第 9 个字,依次记为第三校验字和第四校验字;

[0026] 分别取第一校验字至第四校验字的最高数据组成第一校验输出字,次高数据组成第二校验输出字,次低数据组成第三校验输出字,最低数据组成第四校验输出字;

[0027] 将所述第一校验输出字放置于所述校验矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将所述第二校验输出字放置于所述校验矩阵第 S 行第 9 个数据至第 12 个数据的地址中,将所述第三校验输出字放置于所述校验矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将所述第四校验输出字放置于所述校验矩阵第 S 行第 13 个数据至第 16 个数据的地址中;

[0028] 然后,再顺序从 P1 取出第 16 个字和第 24 个字,依次记为第五校验字和第六校验字,顺序从 P2 取出第 17 个数据和第 25 个数据,依次记为第七校验字和第八校验字;

[0029] 分别取第五校验字至第八校验字的最高数据组成第五校验输出字,次高数据组成第六校验输出字,次低数据组成第七校验输出字,最低数据组成第八校验输出字;

[0030] 将所述第五校验输出字放置于所述校验矩阵第 S+1 行第 1 个数据至第 4 个数据的地址中,将所述第六校验输出字放置于所述校验矩阵第 S+1 行第 9 个数据至第 12 个数据的地址中,将所述第七校验输出字放置于所述校验矩阵第 S+1 行第 5 个数据至第 8 个数据的地址中,将所述第八校验输出字放置于所述校验矩阵第 S+1 行第 13 个数据至第 16 个数据的地址中。

[0031] 当 S 次循环后存在剩余数据时,每次取一个剩余数据放置于所述剩余数据的地址中。

[0032] 当 N 等于 8 进一步包括,根据交织模式填充系统数据的多余比特,根据交织模式填充校验数据的多余比特。

[0033] 从上述技术方案中可以看出,在本发明实施例中,首先根据编码块的数据长度确定交织模式,然后将系统矩阵分为子系统矩阵和将校验矩阵分为子校验矩阵;根据交织模式按列计算系统矩阵地址和校验矩阵地址,将每四个字的系统比特流打包放置于系统矩阵地址中,每四个字的 P1 和每四个字的 P2 打包放置于校验矩阵的地址中,按列输出系统矩阵和校验矩阵。将打包后的数据放置于相对应的地址中,有利于处理器的流水操作,从而能够加快速率匹配的速度。相同的技术方案也可以应用于解速率匹配中,从而能够加快解速率匹配的速度。

附图说明

[0034] 图 1 为现有技术中 Turbo 编码的速率匹配示意图;

[0035] 图 2 为本发明 LTE 系统中 Turbo 编码速率匹配的方法流程示意图;

[0036] 图 3 为本发明实施例中交织模式 1 的示意图;

[0037] 图 4 为本发明实施例中交织模式 2 的示意图;

[0038] 图 5 为本发明实施例中交织模式 3 的示意图;

- [0039] 图 6 为本发明实施例中交织模式 4 的示意图；
[0040] 图 7 为本发明实施例输入数据存储示意图；
[0041] 图 8 为本发明实施例中数据打包操作示意图。

具体实施方式

[0042] 为使本发明的目的、技术方案和优点表达得更加清楚明白，下面结合附图及具体实施例对本发明再作进一步详细的说明。

[0043] 在本发明实施例中，系统比特流、第一校验比特流 (P1) 和第二校验比特流 (P2) 同时并行处理，且连续 4 个字的数据读写操作构造出适合主流处理器实现打包数据操作的数据存取结构，并减少了对内存的读写次数。处理方式简单，循环结构清晰，不存在判断跳转等打断流水线的操作，处理器能够更快的取数据计算，并且把子块交织后的数据直接映射到输出位置上，进而加快速率匹配的速度。

[0044] 在本发明中，Turbo 编码速率匹配包括以下步骤 A 至 E：

[0045] 步骤 A、根据编码块的数据长度 K 确定交织模式。

[0046] 步骤 B、由系统比特流得到系统矩阵，P1 与 P2 交替存放得到校验矩阵，从系统矩阵的第 1 列开始每 8 列为一个子系统矩阵，从校验矩阵的第 1 列开始每 8 列为一个子校验矩阵。

[0047] P1 与 P2 交替存放得到校验矩阵与现有技术的实现方式相同，在此不再赘述。系统矩阵共 32 列，由四个子系统矩阵构成；校验矩阵共 32 列，同样的由四个子校验矩阵构成。

[0048] 步骤 C、依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址，依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址。

[0049] 每次提取子系统矩阵的第 N 列，子校验矩阵的第 N 列，直至完成 8 次循环，以计算系统矩阵中的数据地址和校验矩阵中的数据地址。当 N 小于 8 则继续提取每个子矩阵的第 N+1 列，N 的初始值是 1。

[0050] 步骤 D、选择 4 个系统比特流的字打包后置于所述系统矩阵地址中，选择 4 个 P1 的字和 4 个 P2 的字打包后置于所述校验矩阵的地址中，所述选择包括 S 次循环，S 等于子矩阵的行数 R 减 1 除以 4 向下取整，R 等于 K 加 4 后除以 32 向上取整。

[0051] 对系统比特流的数据打包处理放置于系统矩阵的数据地址中；对 P1 和 P2 的数据打包处理放置于校验矩阵的数据地址中。每个字包括 4 个数据，在本发明的技术方案中，选择四个字即选择 16 个数据。

[0052] 步骤 E、按列输出系统矩阵地址中和校验矩阵地址中的数据，得到速率匹配后的比特流。

[0053] 参见附图 2 是 LTE 系统中 Turbo 编码速率匹配的方法流程示意图，具体包括以下步骤：

[0054] 步骤 201、确定交织模式。

[0055] 由于子块交织矩阵固定为 32 列，根据编码块的数据长度 K 对 32 取余数，余数共有四种情况即 0, 8, 16 和 24，针对于不同的余数需要填充不同的比特。余数是 0 则该编码块属于交织模式 1；余数是 8 则该编码块属于交织模式 2；余数是 16 则该编码块属于交织模式

3 ;余数是 24 则该编码块属于交织模式 4。4 种交织模式决定了矩阵数据地址的偏移量

[0056] LTE 规定了固定的 188 种编码块长度,下面对这 188 种编码块长度进行分析:

[0057] 步长为 8 的编码块长度共计 60 种:

[0058] 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 384, 392, 400, 408, 416, 424, 432, 440, 448, 456, 464, 472, 480, 488, 496, 504, 512。

[0059] 步长为 16 的编码块长度共计 32 种

[0060] 528, 544, 560, 576, 592, 608, 624, 640, 656, 672, 688, 704, 720, 736, 752, 768, 784, 800, 816, 832, 848, 864, 880, 896, 912, 928, 944, 960, 976, 992, 1008, 1024。

[0061] 步长为 32 的编码块长度共计 32 种

[0062] 1056, 1088, 1120, 1152, 1184, 1216, 1248, 1280, 1312, 1344, 1376, 1408, 1440, 1472, 1504, 1536, 1568, 1600, 1632, 1664, 1696, 1728, 1760, 1792, 1824, 1856, 1888, 1920, 1952, 1984, 2016, 2048。

[0063] 步长为 64 的编码块长度共计 64 种

[0064] 2112, 2176, 2240, 2304, 2368, 2432, 2496, 2560, 2624, 2688, 2752, 2816, 2880, 2944, 3008, 3072, 3136, 3200, 3264, 3328, 3392, 3456, 3520, 3584, 3648, 3712, 3776, 3840, 3904, 3968, 4032, 4096, 4160, 4224, 4288, 4352, 4416, 4480, 4544, 4608, 4672, 4736, 4800, 4864, 4928, 4992, 5056, 5120, 5184, 5248, 5312, 5376, 5440, 5504, 5568, 5632, 5696, 5760, 5824, 5888, 5952, 6016, 6080, 6144。

[0065] 其中,对于步长为 32 和 64 的编码块, K 对 32 取余数为 0, 均满足交织模式 1 的要求;步长为 16 的编码块, 528 满足交织模式 3, 544 满足交织模式 1, 依次交替;步长为 8 的编码块中既有满足交织模式 3 的编码块也有满足交织模式 1 的编码块。

[0066] 上述编码块对 32 取余得到每个编码块所属交织模式。

[0067] 交织模式 1 对应的编码块长度: 共计 127 种

[0068] 64, 96, 128, 160, 192, 224, 256, 288, 320, 352, 384, 416, 448, 480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800, 832, 864, 896, 928, 960, 992, 1024, 1056, 1088, 1120, 1152, 1184, 1216, 1248, 1280, 1312, 1344, 1376, 1408, 1440, 1472, 1504, 1536, 1568, 1600, 1632, 1664, 1696, 1728, 1760, 1792, 1824, 1856, 1888, 1920, 1952, 1984, 2016, 2048, 2112, 2176, 2240, 2304, 2368, 2432, 2496, 2560, 2624, 2688, 2752, 2816, 2880, 2944, 3008, 3072, 3136, 3200, 3264, 3328, 3392, 3456, 3520, 3584, 3648, 3712, 3776, 3840, 3904, 3968, 4032, 4096, 4160, 4224, 4288, 4352, 4416, 4480, 4544, 4608, 4672, 4736, 4800, 4864, 4928, 4992, 5056, 5120, 5184, 5248, 5312, 5376, 5440, 5504, 5568, 5632, 5696, 5760, 5824, 5888, 5952, 6016, 6080, 6144。

[0069] 交织模式 2 对应的编码块长度: 共计 15 种

[0070] 40, 72, 104, 136, 168, 200, 232, 264, 296, 328, 360, 392, 424, 456, 488。

[0071] 交织模式 3 对应的编码块长度: 共计 31 种

[0072] 48, 80, 112, 144, 176, 208, 240, 272, 304, 336, 368, 400, 432, 464, 496 ;

[0073] 528, 560, 592, 624, 656, 688, 720, 752, 784, 816, 848, 880, 912, 944, 976, 1008。

[0074] 交织模式 4 对应的编码块长度 : 共计 15 种

[0075] 56, 88, 120, 152, 184, 216, 248, 280, 312, 344, 376, 408, 440, 472, 504。

[0076] 步骤 202、分割系统矩阵和校验矩阵。

[0077] 系统比特流依次组成四个子系统矩阵, 每个子系统矩阵共 8 列, 前 7 列每列有 R 个数据, 第 8 列有 R-1 个数据 ; 校验比特流依次组成四个子校验矩阵, 每个子校验矩阵共 8 列, 前 7 列每列有 2(R-1) 个数据, 第 8 列有 2R 个数据。由于校验矩阵是由 P1 和 P2 交替存放获得的, 因此校验矩阵中每列数据的长度要比系统矩阵每列数据的长度更长。

[0078] 下面对于每种交织模式举例说明 :

[0079] 交织模式 1 : 对应填充比特 $N_b = 28$, 即比特流的长度中的编码块长度可以被 32 整除的情况。例 K = 6144, 参见附图 3。

[0080] 前 32 列较短的长度对应系统矩阵, 后 32 列较长的长度对应校验矩阵。系统比特依次由四个子系统矩阵构成, 其中短的矩形条是不含多余比特的列, 长度为 R-1, 长的矩形条是含多余比特的列, 长度为 R。多余比特是交织矩阵中不满足列长度多余的比特。校验比特依次由四个子校验矩阵构成, 其中短的矩形条是不含多余比特的列, 长度为 2(R-1), 长的矩形条是含多余比特的列, 长度为 2R。

[0081] 交织模式 2 : 对应 $N_b = 20$, 即比特流的长度中的编码块长度对 32 取余为 8 的情况。例 K = 488, 参见附图 4。

[0082] 交织模式 3 : 对应 $N_b = 12$, 即比特流的长度中的编码块长度对 32 取余为 16 的情况。例 K = 496, 参见附图 5。

[0083] 交织模式 4 : 对应 $N_b = 4$, 即比特流的长度中的编码块长度对 32 取余为 32 的情况。例 K = 504, 参见附图 6。

[0084] 类似于交织模式 1, 交织模式 2、3、4 与交织模式 1 不同点在于 K 的不同, 其系统矩阵与校验矩阵相应的列长度不同。

[0085] 步骤 203、计算第 N 列的系统矩阵地址和校验矩阵地址。

[0086] 依次提取每个子系统矩阵的第 N 列后根据交织模式计算该列数据对应的系统矩阵地址, 依次提取每个子校验矩阵的第 N 列后根据交织模式计算该列数据对应的校验矩阵地址。

[0087] 每个子系统矩阵共 8 列, 一次循环计算每个子系统矩阵的一列数据地址, 共进行八次循环就可以计算出所有系统矩阵地址 ; 相应的, 每个子校验矩阵也是 8 列, 一次循环计算每个子校验矩阵的一列数据地址, 共进行八次循环就可以计算出所有校验矩阵地址。因此 N 的初始值是 1, 且 N 的最大值是 8。

[0088] 步骤 2031、计算交织索引。

[0089] 交织索引是根据交织模式所确定的矩阵中每个数据地址的偏移量的列表。针对不同的交织模式, 计算出每种交织模式中每个数据地址的偏移量, 相应的系统矩阵中每个数据地址增加相应的偏移量得到中间偏移地址, 然后根据交织模式和 N 整体偏移每个数据的中间偏移地址得到系统矩阵地址 ; 校验矩阵中每个数据地址增加相应的偏移量得到中间偏移地址, 然后根据交织模式和 N 整体偏移每个数据的中间偏移地址得到校验矩阵地址。

[0090] 下面详细说明交织索引的计算 :

[0091] 首先根据冗余版本参数 RV 和速率匹配软 Buffer N_{cb} 的大小, 以及 LTE 协议描述的

公式计算出起始列 k_0 :

[0092]

$$k_0 = R \cdot \left(2 \cdot \left\lceil \frac{N_{cb}}{8R} \right\rceil \cdot RV + 2 \right) \quad (1)$$

[0093] 上式中, R 为子系统矩阵的行数, N_{cb} 为速率匹配软 Buffer 大小, RV 为冗余版本参数, 范围 :0, 1, 2, 3。其中, N_{cb} 为已知参数, RV 是已知参数。

[0094] 另外, 由于矩阵排列形式的特殊性, 如果 k_0 大于 32, 则系统矩阵的起始列位置等于 k'_0 :

$$k'_0 = \frac{k_0}{2} + 16 \quad (2)$$

[0096] 公式 (2) 保证了 k'_0 的取值范围是 2 到 53。由于校验矩阵是 P1 和 P2 合并在一起, 如果校验矩阵每列数据数目与系统矩阵每列数据相同, 则校验矩阵应该是 64 列。而对于 LTE 系统而言, P1 和 P2 单独计算, 则校验矩阵共 96 列。如果按照原公式 (1) 计算, 则 k_0 的取值范围为 2 到 74。通过公式 (2) 计算后, 相当于把原来对应的列数折算成校验矩阵所在的列数。例如原来是 74 列, 经过公式 (2) 计算后是在第 53 列。

[0097] 系统矩阵的起始列位置确定后, 根据四种交织模式分别计算交织索引。

[0098] 对每一列的长度赋值即初始化每一列的长度, 赋值可能为 R , $R-1$ 或者 $2R$, $2(R-1)$ 共计四种可能。

[0099] 第 K 列的开头位置的输出索引值为 0, 第 $K+1$ 列的开头位置的输出索引值是第 K 列开头位置的输出索引值加上第 K 列的长度, 以此类推。特别的, 由于 RV 参数的原因, 第 0 列始终会放到后面, 因此第 0 列的开头位置的输出索引值是第 63 列的开头位置的输出索引值加上第 63 列的长度。

[0100] 此外, 还需要计算多余比特的输出地址索引, 对于系统矩阵, 由于仅有 1 个比特的多余比特, 多余比特的输出地址相当于该多余比特所在列的下一列的输出地址索引减 1; 对于校验矩阵, 由于有 2 个比特的多余比特, 多余比特的输出地址相当于该多余比特所在列的下一列的输出地址索引减 2。

[0101] 结合填充比特的个数, 可以得到如下 4 种交织模式的特点。

[0102] 交织模式 1, $N_b = 28$, 系统比特和 P1 的交织模式如下 :

[0103] $\langle 4, 20, 12, 28, 8, 24, 16, 0, 6, 22, 14, 30, 10, 26, 18, 2, 5, 21, 13, 29, 9, 25, 17, 1, 7, 23, 15, 31, 11, 27, 19, 3 \rangle$

[0104] 当系统比特流和 P1 取第 31 个元素的时候, 即图 3 矩阵中的第 59 列, P2 取第 0 个元素。这个时候取数据有一个跳变, 因此, 为了构造特殊的结构, 需要把我们定义的矩阵中的 59 到 64 列的输出列的起始地址依次向后偏移 1 个单位, 同时把相应列的多余比特起始地址向后偏移 1 个单位。

[0105] 交织模式 2, $N_b = 20$, 系统比特和 P1 的交织模式如下 :

[0106] $\langle 12, 28, 20, 4, 16, 0, 24, 8, 14, 30, 22, 6, 18, 2, 26, 10, 13, 29, 21, 5, 17, 1, 25, 9, 15, 31, 23, 7, 19, 3, 27, 11 \rangle$

[0107] 当系统比特流和 P1 取第 31 个元素的时候, 即图 4 矩阵中的第 57 列, P2 取第 0 个元素。这个时候取数据有一个跳变, 因此, 为了构造特殊的结构, 需要把我们定义的矩阵中

的 57 到 64 列的输出列的起始地址依次向后偏移 1 个单位,同时把相应列的多余比特起始地址向后偏移 1 个单位。

[0108] 交织模式 3, $N_b = 12$, 系统比特和 P1 的交织模式如下:

[0109] $\langle 20, 4, 28, 12, 24, 8, 0, 16, 22, 6, 30, 14, 26, 10, 2, 18, 21, 5, 29, 13, 25, 9, 1, 17, 23, 7, 31, 15, 27, 11, 3, 19 \rangle$

[0110] 当系统比特流和 P1 取第 31 个元素的时候,即图 5 矩阵中的第 58 列,P2 取第 0 个元素。这个时候取数据有一个跳变,因此,为了构造特殊的结构,需要把我们定义的矩阵中的 58 到 64 列的输出列的起始地址依次向后偏移 1 个单位,同时把相应列的多余比特起始地址向后偏移 1 个单位。

[0111] 交织模式 4 的, $N_b = 4$, 系统比特和 P 1 的交织模式如下:

[0112] $\langle 28, 12, 4, 20, 0, 16, 8, 24, 30, 14, 6, 22, 2, 18, 10, 26, 29, 13, 5, 21, 1, 17, 9, 25, 31, 15, 7, 23, 3, 19, 11, 27 \rangle$

[0113] 当系统比特流和 P1 取第 31 个元素的时候,即矩阵中的第 56 列,P2 取第 0 个元素。此时取数据有一个跳变,因此,为了构造特殊的结构,需要将矩阵中的 56 到 64 列输出列的起始地址依次向后偏移 1 个单位,同时把相应列的多余比特起始地址向后偏移 1 个单位。

[0114] 交织索引的大小是 $64+56 = 120$ 个长度。前 64 个索引存储系统矩阵 32 列、校验矩阵 32 列共计 64 列的起始地址索引,即开头的比特在输出缓存中的位置;后 56 个索引是每个特殊的长度列的多余比特在输出缓存中的位置。由于校验矩阵的含多余比特的长列中两个多余比特是紧密连接的,因此只需要计算一个地址索引即可,另外一个多余比特的地址可以根据该地址索引加

[0115] 交织模式 1 的系统矩阵和校验矩阵分别有 4 个多余比特在最后一行,这里之所以是最后一行是把填充比特 NULL 放到了最后一行,因此称为多余比特,仅需要 8 个多余比特的索引。同样地,交织模式 2 需要 24 个剩余比特的索引。交织模式 3 需要 40 个剩余比特的索引,交织模式 4 需要 56 个剩余比特的索引。按照交织模式 4 取最大的长度是 56,上述共计 120 个索引。这 120 个地址索引,决定了输出数据存储的偏移量。

[0116] 在速率匹配之前预先计算好上述索引参数。由于 LTE 系统存在多个编码块的长度相同的情况,因此对于相同的编码块仅需要计算一次交织索引即可。

[0117] 步骤 2032、根据交织模式和 N 整体偏移中间偏移地址分别得到系统矩阵地址和校验矩阵地址。

[0118] 由于子块交织器中的字块交织矩阵有 32 列,则编码块的填充比特等于 32 减去 $K+4$ 对 32 取余。系统比特流、P1 和 P2 的长度均等于 $K+4$ 。即:交织模式 1 对应的填充比特 N_b 等于 28 个;交织模式 2 对应的 N_b 等于 20 个;交织模式 3 对应的 N_b 等于 12 个;交织模式 4 对应的 N_b 等于 4 个。

[0119] 根据 N_b 结合表一中 P 1 的交织模式的前八个数据,可以得到四种交织模式每一列数据的整体偏移量。当 32 减去 N_b 后,与 P1 第 N 个数据的和小于等于 32,整体偏移量 H 等于 32 减去 N_b 后与 P1 第 N 个数据的和;当 32 减去 N_b 后,与 P1 的第 N 个数据的和大于 32,整体偏移量 H 等于 P1 的第 N 个数据减去 N_b 。

[0120] 表 1 P1 校验矩阵交织模式

[0121] $\langle 0, 16, 8, 24, 4, 20, 12, 28, 2, 18, 10, 26, 6, 22, 14, 30, 1, 17, 9, 25, 5, 21, 13, 29, 3,$

19, 11, 27, 7, 23, 15, 31>

[0122] 交织模式 1 : $N_b = 28$ 对应每一列的 H 分别为 :4, 20, 12, 28, 8, 24, 16, 0 ;

[0123] 交织模式 2 : $N_b = 20$ 对应每一列的 H 分别为 :12, 28, 20, 4, 16, 0, 24, 8 ;

[0124] 交织模式 3 : $N_b = 12$ 对应每一列的 H 分别为 :20, 4, 28, 12, 24, 8, 0, 16 ;

[0125] 交织模式 4 : $N_b = 4$ 对应每一列的 H 分别为 :28, 12, 4, 20, 0, 16, 8, 24。

[0126] 即对于交织模式 1 的编码块,其子系统矩阵和子校验矩阵的第一列的整体偏移量是 4,第二列的整体偏移量是 20,以此类推可以得到第三列至第八列的整体偏移量。本文中的整体偏移量是指该列数据的地址的偏移量。

[0127] 步骤 204、循环 S 次打包系统比特流的数据,打包 P1 和 P2 的数据。

[0128] 在步骤 203 中已经计算得到系统矩阵和校验矩阵中数据对应的地址。步骤 204 中将系统比特流的数据打包放置于系统矩阵中的地址 ;将 P1 和 P2 的数据打包放置于校验矩阵中的地址。循环次数 S 等于子系统矩阵的行数 R 减 1 除以 4 向下取整, R 等于 K 加 4 后除以 32 向上取整。

[0129] 当 S 次循环后若存在剩余数据,则每次取一个剩余数据放置于剩余数据的地址中。

[0130] 对于系统比特流、P1 和 P2 的输入数据共三个码流的存放如附图 7 所示。在系统比特流中的第一个字由数据 S_0 、数据 S_1 、数据 S_2 和数据 S_3 构成,P1 的第一个字由数据 P_0 、数据 P_1 、数据 P_2 和数据 P_3 构成,而 P2 的第一个字仅由在第四个位置中的数据 B_0 构成。若系统比特流、P1 和 P2 的输入数据不满足上述条件,则需要将不满足条件的比特流调整到上述条件。调整方式是现有技术,在此就不再赘述。

[0131] 下面详细介绍将系统比特流的字打包放置于系统矩阵中的地址。

[0132] 从第 0 个字开始,每隔 8 个字取出系统比特流的字,得到第一系统字 A_0 、第二系统字 A_8 、第三系统字 A_{16} 和第四系统字 A_{24} 。参见附图 8,分别取四个系统字的最高数据组成第一系统输出字 B_0 ,次高数据组成第二系统输出字 B_8 ,次低数据组成第三系统输出字 B_{16} ,最低数据组成第四系统输出字 B_{24} 。上述过程为数据打包过程。其中,一个字是由四个数据构成。

[0133] 将 B_0 放置于系统矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将 B_8 放置于系统矩阵第 S 行第 9 个数据至第 12 个数据的地址中,将 B_{16} 放置于系统矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将 B_{24} 放置于系统矩阵第 S 行第四列的第 13 个数据至第 16 个数据中。

[0134] 进行 S 次循环后,系统比特流中的字分别放置于系统矩阵中的地址中。

[0135] 将 P1 和 P2 的数据打包放置于校验矩阵中的地址,与对系统矩阵的数据打包不同之处在于,由于校验矩阵中的数据数目是系统矩阵中数据数目的两倍,因此系统矩阵进行一次数据打包,相应的校验矩阵进行二次数据打包。

[0136] 顺序从 P1 取出第 0 个字和第 8 个字,依次记为第一校验字和第二校验字,顺序从 P2 取出第 1 个字和第 9 个字,依次记为第三校验字和第四校验字。

[0137] 分别取第一校验字至第四校验字的最高数据组成第一校验输出字,次高数据组成第二校验输出字,次低数据组成第三校验输出字,最低数据组成第四校验输出字。

[0138] 将第一校验输出字放置于校验矩阵第 S 行第 1 个数据至第 4 个数据的地址中,将

第二校验输出字放置于校验矩阵第 S 行第 9 个数据至第 12 个数据的地址中,将第三校验输出字放置于校验矩阵第 S 行第 5 个数据至第 8 个数据的地址中,将第四校验输出字放置于所述校验矩阵第 S 行第 13 个数据至第 16 个数据的地址中。

[0139] 然后,再顺序从 P1 取出第 16 个字和第 24 个字,依次记为第五校验字和第六校验字,顺序从 P2 取出第 17 个字和第 25 个字,依次记为第七校验字和第八校验字。

[0140] 分别取第五校验字至第八校验字的最高数据组成第五校验输出字,次高数据组成第六校验输出字,次低数据组成第七校验输出字,最低数据组成第八校验输出字。

[0141] 将第五校验输出字放置于校验矩阵第 S+1 行第 1 个数据至第 4 个数据的地址中,将第六校验输出字放置于校验矩阵第 S+1 行第 9 个数据至第 12 个数据的地址中,将第七校验输出字放置于校验矩阵第 S+1 行第 5 个数据至第 8 个数据的地址中,将第八校验输出字放置于校验矩阵第 S+1 行第 13 个数据至第 16 个数据的地址中。

[0142] 步骤 205、判断 N 小于 8。

[0143] 判断 N 是否小于等于 8,若 N 小于 8,则令 N+1 后返回步骤 103 ;否则,执行步骤 106。

[0144] 由于系统矩阵由四个子系统矩阵构成,校验矩阵由四个子校验矩阵构成。子系统矩阵和子校验矩阵均有 8 列数据,当 $N = 8$ 即子系统矩阵中每列数据均放置于系统矩阵的地址中,以及子校验矩阵中每列数据均放置于校验矩阵的地址中。

[0145] 至此,系统矩阵地址中的数据和校验矩阵地址中的数据是速率匹配后的数据。

[0146] 步骤 206、按列输出系统矩阵地址中的数据和校验矩阵地址中的数据。

[0147] 按列从第 1 列至第 32 列,依次输出系统矩阵地址中的数据 ;然后,按列从第 33 列至第 64 列,依次输出校验矩阵地址中的数据,得到速率匹配后的数据。

[0148] 另外,步骤 201 至步骤 206 的技术方案还适应于 LTE 系统中 Turbo 编码解速率匹配。其技术方案的实现过程与 Turbo 编码速率匹配相同,不同点在于步骤 2031 中,起始列的位置等于 0。

[0149] 以上所述,仅为本发明的较佳实施例而已,并非用于限定本发明的保护范围。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

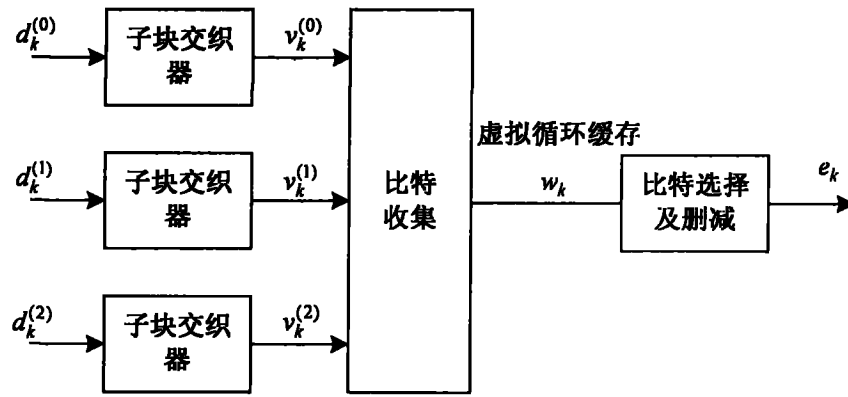


图 1

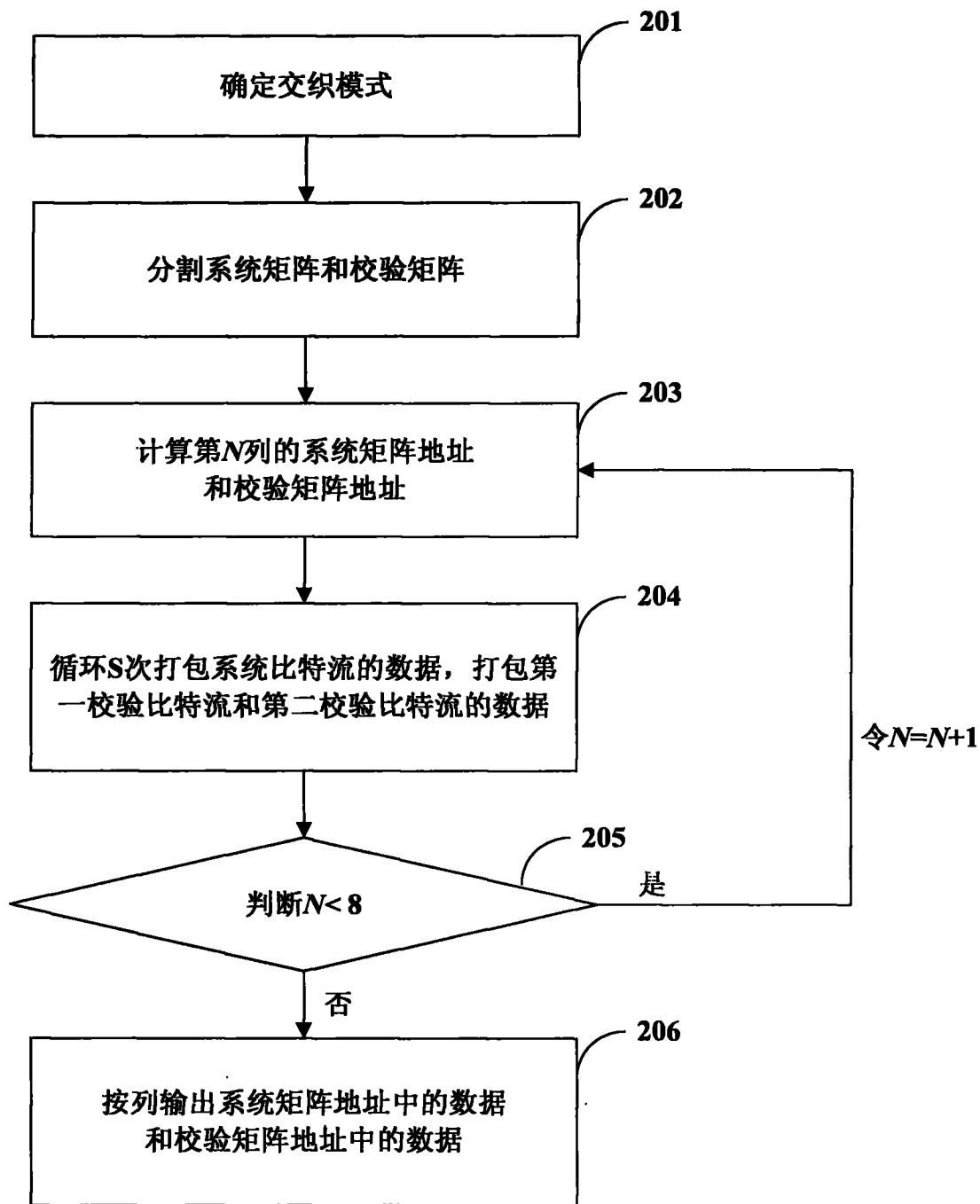


图 2



图 3

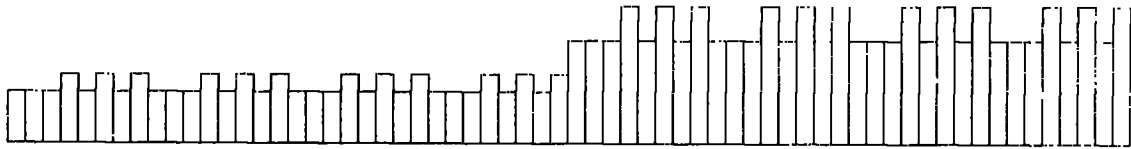


图 4

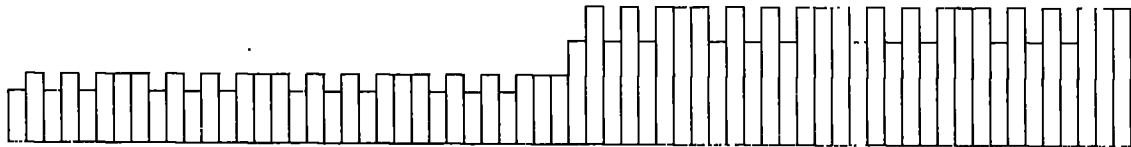


图 5

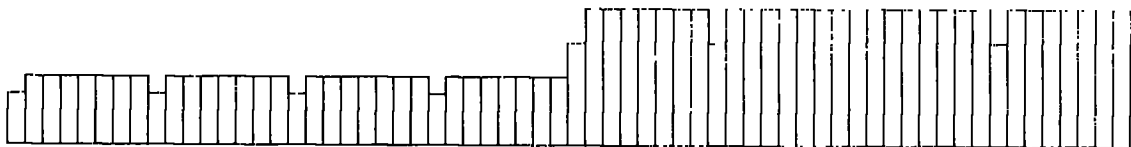


图 6

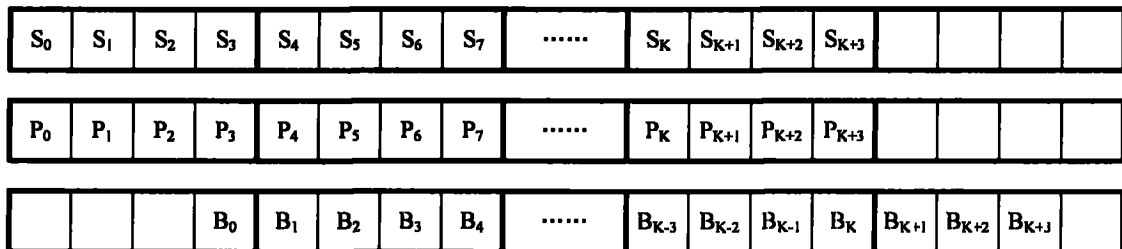


图 7

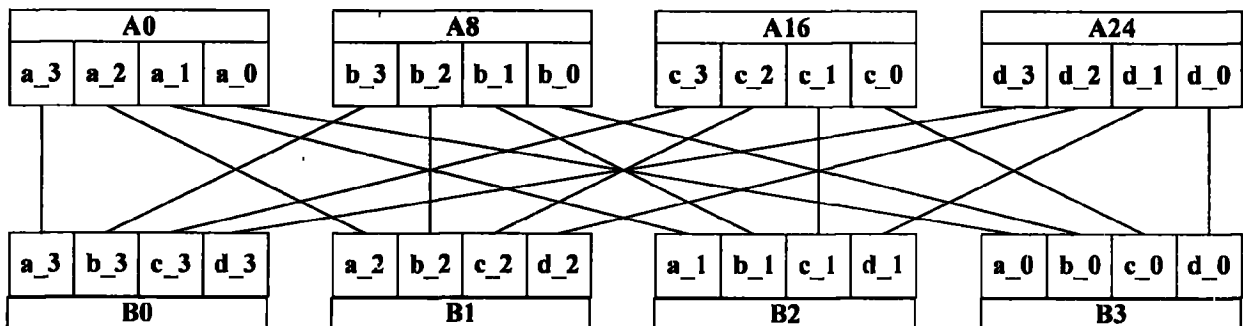


图 8