

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2018 年 5 月 24 日 (24.05.2018)



(10) 国际公布号
WO 2018/090629 A1

(51) 国际专利分类号:
H04L 1/00 (2006.01)

(21) 国际申请号: PCT/CN2017/089808

(22) 国际申请日: 2017 年 6 月 23 日 (23.06.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201611022892.0 2016 年 11 月 21 日 (21.11.2016) CN

(71) 申请人: 深圳市中兴微电子技术有限公司
(SANECHIPS TECHNOLOGY CO., LTD.) [CN/CN];
中国广东省深圳市南山区西丽街道留仙大道
中兴工业园, Guangdong 518055 (CN)。

(72) 发明人: 徐心明(XU, Xinming); 中国广东省深圳市
南山区西丽街道留仙大道中兴工业园, Guangdong
518055 (CN)。 陈云生(CHEN, Yunsheng); 中国
广东省深圳市南山区西丽街道留仙大道中
兴工业园, Guangdong 518055 (CN)。

(74) 代理人: 北京派特恩知识产权代理有
限公司(CHINA PAT INTELLECTUAL PROPERTY
OFFICE); 中国北京市海淀区海淀南路21号中关
村知识产权大厦B座2层, Beijing 100080 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家
保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,
BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU,
CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS,

(54) Title: METHOD OF DEINTERLEAVING AND DE-RATE MATCHING, DEVICE, AND COMPUTER STORAGE MEDIUM

(54) 发明名称: 一种解交织解速率匹配的方法、装置及计算机存储介质

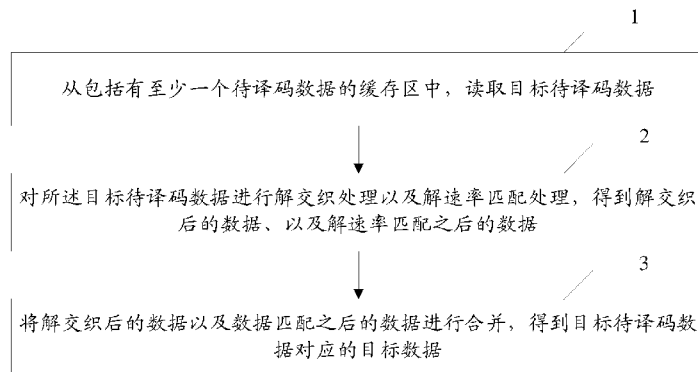


图 1-1

- 1 Read, from a buffer including at least one data to be decoded, target data to be decoded
- 2 Perform a deinterleaving process and a de-rate matching process on the target data to be decoded to obtain deinterleaved data, and de-rate matched data
- 3 Combine the deinterleaved data and the de-rate matched data to obtain the target data corresponding to the target data to be decoded

(57) Abstract: The invention discloses a method of deinterleaving and de-rate matching, a device, and a computer storage medium. The method comprises: reading, from a buffer including at least one data to be decoded, target data to be decoded; performing a deinterleaving process and a de-rate matching process on the target data to be decoded to obtain deinterleaved data, and de-rate matched data; and combining the deinterleaved data and the de-rate matched data to obtain the target data corresponding to the target data to be decoded.

(57) 摘要: 本申请公开了一种解交织解速率匹配的方法、装置及计算机存储介质, 方法包括: 从包括有至少一个待译码数据的缓存区中, 读取目标待译码数据; 对所述目标待译码数据进行解交织处理以及解速率匹配处理, 得到解交织后的数据、以及解速率匹配之后的数据; 将解交织后的数据以及数据匹配之后的数据进行合并, 得到目标待译码数据对应的目标数据。



WO 2018/090629 A1

JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告 (条约第21条(3))。

一种解交织解速率匹配的方法、装置及计算机存储介质

相关申请的交叉引用

本申请基于申请号为 201611022892.0、申请日为 2016 年 11 月 21 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此引入本申请作为参考。

技术领域

本申请涉及通信领域，尤其涉及一种解交织解速率匹配的方法、装置及计算机存储介质。

背景技术

长期演进(Long Term Evolution, LTE)是由第三代合作伙伴计划(The 3rd Generation Partnership Project, 3GPP)组织制定的通用移动通信系统(Universal Mobile Telecommunications System, UMTS)技术标准的长期演进。LTE 系统引入了正交频分复用(Orthogonal Frequency Division Multiplexing, OFDM)和多输入多输出(Multi-Input & Multi-Output, MIMO)等关键技术，显著增加了频谱效率和数据传输速率。

随着 LTE 的业务和 LTE 终端的不断发展，需要处理的数据流量也成倍增加，LTE 采用了 Turbo 编码，速率匹配和 HARQ 重传的先进的机制来保证数据业务的 QOS。在 LTE 的数据信道的编码过程中，数据信道在 MAC 层经过处理以后，在物理层经过码块分割，码块 CRC 添加，TB CRC 添加，Turbo 编码，交织，速率匹配等符号级处理的步骤之后，在经过 FFT 模块生产 OFDM 符号发给空口。

目前，解交织和解速率的方法主要是解交织解速率匹配的并行度较低，

采用 8 比特或者 4 比特并行处理，硬件实现面积和功耗较大，目前的解交织和解速率匹配的仅仅适用于基站或者 UE 侧。

发明内容

有鉴于此，本申请的目的在于提供一种解交织解速率匹配的方法、装置及计算机存储介质，能至少解决现有技术中存在的上述问题。

为达到上述目的，本申请的技术方案是这样实现的：

本申请提供了一种解交织解速率匹配的方法，所述方法包括：

从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

对所述目标待译码数据进行解交织处理以及解速率匹配处理，得到解交织后的数据、以及解速率匹配之后的数据；

将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

本申请提供了一种解交织解速率匹配的装置，所述装置包括：

数据缓存单元，用于缓存至少一个待译码数据；

解交织解速率匹配控制单元，用于从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

数据解交织单元，用于对所述目标待译码数据进行解交织处理，得到解交织后的数据；

速率匹配单元，用于对所述目标待译码数据进行解速率匹配处理，得到解速率匹配之后的数据；

解交织解速率匹配数据缓存单元，用于缓存解速率匹配以及解交织后的数据；

合并单元，用于将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

本申请还提供了一种计算机存储介质，该计算机存储介质包括一组计

计算机可执行指令，所述计算机可执行指令配置为执行前述方法的步骤。

本申请还提供了一种解交织解速率匹配的装置，包括：处理器和用于存储能够在处理器上运行的计算机程序的存储器，

其中，所述处理器用于运行所述计算机程序时，执行前述方法的步骤。

- 5 本申请提供了一种解交织解速率匹配的方法、装置及计算机存储介质，采用高并行处理的解交织解速率匹配的方法，可以成倍的提高处理的能力，减少处理延迟。

附图说明

图 1-1 为本申请解交织解速率匹配的方法流程示意图；

- 10 图 1-2 为本申请解交织解速率匹配的装置组成结构示意图一；

图 1-3 为本申请解交织解速率匹配的装置组成结构示意图二；

图 2 为本申请解交织解速率匹配的装置组成结构示意图三。

具体实施方式

- 15 本申请的核心思想是：先从 TB 数据待译码的缓存区中读取 CB 块的译码数据缓存在 CB 块的译码缓存区之中，然后对 CB 译码缓存区的数据进行解交织，同时从速率匹配的数据缓存中把对应的速率匹配数据读出来和解交织后的数据进行合并，完成解交织和速率匹配的过程。

下面结合附图及具体实施例对本申请再作进一步详细的说明。

- 20 本申请提供一种解交织解速率匹配的方法，如图 1-1 所示，所述方法包括：

步骤 1：从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

步骤 2：对所述目标待译码数据进行解交织处理以及解速率匹配处

理，得到解交织后的数据、以及解速率匹配之后的数据；

步骤 3：将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

所述对所述目标待译码数据进行解交织处理以及解速率匹配处理之前，所述方法还包括：

计算得到所述解交织处理以及解速率匹配处理所需要的参数。

所述计算得到所述解交织处理以及解速率匹配处理所需要的参数，包括以下至少之一：

计算交织矩阵的行数 R ；计算出交织矩阵的填充比特个数 nd ；计算起始位置 $k0$ 所对应的类型标识 $k0_flag$ ；计算 $k0$ 在矩阵的列号 $k0_col$ ；计算 $k0$ 在矩阵的列号 $k0_row$ ；速率匹配的数据的结束位置 ncb 所对应的类型标识 ncb_flag ；计算 ncb 的在矩阵的列号 ncb_col ；计算 ncb 在矩阵的行号 ncb_row 。

解交织解速率匹配所有的参数，可以参见表 1：

参数名称	具体的参数含义	参数的取值范围	备注
$cbsize$	CB 码块的大小	40-6144	协议规定的 188 种值
e	解交织解速率匹配的输入数据长度		没有固定的取值范围
$k0$	速率匹配的数据的起始位置	0-18527	在 3GPP 212 协议中规定了计算方法
ncb	速率匹配的数据的结束位置	0-18527	在 3GPP 212 协议中规定了计

			算方法
<i>R</i>	速率匹配的交织 行数	2-193	
<i>nd</i>	交织矩阵第 0 行 需要填充的无效 比特个数	4,12,20,28	
<i>k0_flag</i>	<i>k0</i> 速率匹配起 始位置是属于系 统比特还是校验 比特	0-1	0: 代表系统比 特, 1: 代表校 验比特
<i>k0_row</i>	<i>k0</i> 速率匹配起 始位置在交织矩 中的行号	0-192	
<i>k0_col</i>	<i>k0</i> 速率匹配起 始位置在交织矩 中的列号	0-31	
<i>ncb_flag</i>	<i>ncb</i> 速率匹配结 束位置是属于系 统比特还是校验 比特	0-1	0: 代表系统比 特, 1: 代表校 验比特
<i>ncb_row</i>	<i>ncb</i> 速率匹配结 束位置在交织矩 中的行号	0-192	
<i>ncb_col</i>	<i>ncb</i> 速率匹配结 束位置在交织矩	0-31	

	中的行号		
<i>row_cnt</i>	解交织解速率匹配过程中的行计数器	0-192	
<i>col_cnt</i>	解交织解速率匹配过程中的列计数器	0-31	
<i>col_cnt_write</i>	解交织解速率匹配的过程中的实际写列计数器	0-31	
<i>process_flag</i>	解交织解速率匹配的过程中是属于系统比特还是校验比特	0-1	0: 代表系统比特, 1: 代表校验比特

表 1

本申请提供了一种解交织解速率匹配的方法，详细的处理过程：

首先，整个传输块（Transport Block，TB）的数据缓存区，一般选择存储容量大的 L2D 或者 DDR；可以采用先入先出（First Input First Output，FIFO）或者 RAM 实现，LTE 的 *cbsize* 大小从 40 到 6144，经过编码后速率匹配的长度是 e ，FIFO 的缓存区的大小可以设置为 e 个 byte，一般在译码的时候用 1 个 byte 来表示软信息大小，控制模块在 FIFO 有空间的时候，就把外部的 TB 数据缓存到 CB 数据缓存区。

接收外部配置的 TB 任务，包括关键的参数 $k0$ ， ncb ， $cbsize$ ， e ， $k0$ 在速率匹配的过程之中是数据编码的开始位置， ncb 是速率匹配过程之中

数据编码的卷绕位置, 如果到了 ncb 卷绕位置, 数据长度还未达到 e , 则编码重新从 0 开始取数据。Cbsize 是 LTE 编码的 CB 的码块长度, 是固定的 188 种长度, 范围是 40 到 6144, e 是代表数据译码的长度, 一般单位是 byte。

5 然后按照下面步骤计算出解交织解速率匹配需要的参数和解交织的 RAM 初始化: 其中, 所述初始化就是把 RAM 的所有的地址的数据初始化为全 0。具体的:

第一步: 计算交织矩阵的行数 R , $R = cbsize / 32 + 1$,

第二步: 计算出交织矩阵的填充比特个数 nd , $nd = 28 - cbsize[4:0]$ 。

10 第三步: 计算 $k0$ 的起始位置在矩阵是属于系统比特还是校验比特 $k0_flag$, 如果 $k0 < 32 * R$, $k0_flag = 0$, 否则 $k0_flag = 1$ 。

第四步: 计算 $k0$ 的起始位置在矩阵的列号 $k0_col$, 如果 $k0 < 32 * R$, 则 $k0_col = \lfloor k0 / R \rfloor$, 否则 $k0_col = \lfloor (k0 - 32 * R) / 2 / R \rfloor$ 。

第五步: 计算 $k0$ 的起始位置在矩阵的列号 $k0_row$, 如果 $k0 < 32 * R$,
15 则 $k0_row = k0 - \lfloor k0 / R \rfloor * R$, 否则 $k0_row = (k0 - 32 * R) / 2 - k0_col * R$ 。

第六步: ncb 的结束位置在矩阵是属于系统比特还是校验比特 ncb_flag , 如果 $ncb < 32 * R$, $ncb_flag = 0$, 否则 $ncb_flag = 1$ 。

第七步: 计算 ncb 的结束位置在矩阵的列号 ncb_col , 如果 $ncb < 32 * R$, 则 $ncb_col = \lfloor ncb / R \rfloor$, 否则 $ncb_col = \lfloor (ncb - 32 * R) / 2 / R \rfloor$ 。

20 第八步: 计算 ncb 的结束位置在矩阵的行号 ncb_row , 如果 $ncb < 32 * R$, 则 $ncb_row = ncb - \lfloor ncb / R \rfloor * R$, 否则 $ncb_row = (ncb - 32 * R) / 2 - ncb_col * R$ 。

第九步: CB 解交织解速率匹配控制 203 会发出 CB 解交织解速率匹配缓存 212 的所有写地址进行写操作, 把该缓存区都写成 0, 这样后面速率匹配的过程不管是打孔模式的速率匹配, 还是重复模式的速率匹配,

还是多个超级小区合并的解速率匹配,都统一到一种硬件处理单元之中。

在解速率的过程之中,设置 row_cnt 和 col_cnt 2 个变量,初始值是 $k0_row$ 和 $k0_col$, 还设置 1 个 $process_flag$, 初始值就是 $k0_flag$ 。

然后,从 CB 缓存区中顺序读取数据,下面都以 $cbsize$ 为 6144 的码
5 块长度来说明解交织解速率匹配的长度, $cbsize$ 为 6144 的交织矩阵的行数是 193 行,读地址产生模块每个 cycle 从 CB 缓存区 202 中读取 128 比特数据或者 256 比特数据,也就是 16 或者 32 个待解交织的软符号。

用寄存器来缓存从 CB 数据缓存区读出的 128 比特或者 256 比特数据。

10 所述对所述目标待译码数据进行解速率匹配处理,包括:

目标待译码数据的至少一列数据中分别提取每一列数据;其中,所述目标待译码数据为矩阵数据;

针对提取到的每一列数据,根据所述列数据中每一个数据对应的行数,以及每一个数据对应的类型,确定增加的数值;

15 基于每一个数据对应的增加的数值,对所述目标待译码数据进行解速率匹配。

具体来说,解交织解速率匹配的过程是按列来解交织解速率匹配的,由于编码过程之中进行了列交换,在译码过程之中也要进行相应的列交换,交换的方法如下表 2 所示:

列	0	1	2	3	4	5	6	7
实际列	0	16	8	24	4	20	12	28
列	8	9	10	11	12	13	14	15
实际列	2	18	10	26	6	22	14	30
列	16	17	18	19	20	21	22	23
实际列	1	17	9	25	5	21	13	29

列	24	25	26	27	28	29	30	31
实际列	3	19	11	27	7	23	15	31

表 2

解速率匹配的过程是按照列来进行，在某列速率匹配的过程之中， col_cnt 保持不变，而 row_cnt 在系统比特解交织解速率匹配的系统比特的时候固定增加 16 或者 32；而在校验比特的时候， row_cnt 固定增加 8 或者 16。上述 16 和 32 对应 2 种并行度的处理方式。实际应用中可以根据需求采用其中的一种或者更高并行度的处理。

写地址产生的规则就是 $col_cnt_write[0] + row_cnt[7:4]$ （16 比特并行处理）或者 $col_cnt_write[0] + row_cnt[7:5]$ （32 比特并行处理）。

所述对所述目标待译码数据进行解交织处理，包括：

10 目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；另外，需要指出的是，矩阵列数是固定的，行数可以不固定；

从所述每一列数据中提取得到每一行数据，判断所述每一行数据为系统比特或校验比特，基于所述判断结果对所述数据进行解交织。

15 具体来说，对数据进行解交织处理，分成 2 种情况

第一种：当所述判断结果表征所述数据为系统比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置填充为 0；若所述数据处于第 N 行，直接将读取到的数据作为解交织后的数据；其中，N 为大于等于 1 的整数。 $process_flag$ 为 0，代表当前处理的数据是系统比特，解交织和解速率匹配的过程主要有 3 步骤：

第一步： $row_cnt = 0$ 的时候，也就是在进行第零行的处理时，首先判断 col_cnt_write 是否小于 nd （交织矩阵第 0 行需要填充的无效比特个数），

如果 $col_cnt_write < nd$ ，要那么在数据前面填充 1 个 byte 的 0，否则不用填充 1 个 byte 的 0；然后从填充后的数据中读取 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理）。

第二步： row_cnt 不是最后一行的情况，将读取的数据 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理）直接做为解交织后的数据。

第三步： row_cnt 是最后一行的时候，真实有效的数据不够 128 比特或者 256 比特，需要填充比特。填充的比特个数是 $16 - R[3:0]$ （16 比特并行处理）或者 $32 - R[4:0]$ （32 比特并行处理）个 0，然后把填充后的数据作为真实的写数据。

第二种：当所述判断结果表征所述数据为校验比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置之前的指定比特数填充为 0；

若所述数据处于第 N 行、且所述数据不处于最后行，将读取的数据 128 比特或者 256 比特，按照奇偶分别产生 64 比特或者 128 比特的校验比特数据；

若所述数据处于最后行，则确定所处行中对应的填充比特以及填充比特数量。 $process_flag$ 为 1，代表当前处理的数据是校验比特，解交织和解速率匹配的过程主要有 3 步骤：

第一步： $row_cnt = 0$ 的时候，判断 col_cnt_write 是否大于 nd ，如果 $col_cnt_write < nd$ ，要在数据前面填充 2 个 byte 的 0，否则不用填充 2 个 byte 的 0。然后从填充后的数据中读取 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理），按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特（16 比特并行处理）或者 128 比特（32 比特并行处理）的校验比特 1 和校验比特 2 数据。

第二步: row_cnt 不是最后一行的情况, 将读取的数据 128 比特 (16 比特并行处理) 或者 256 比特 (32 比特并行处理), 按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特 (16 比特并行处理) 或者 128 比特 (32 比特并行处理) 的校验比特 1 和校验比特 2 数据。

5 第三步: row_cnt 是最后一行的时候, 真实有效的数据不够 128 比特或者 256 比特, 需要填充比特。填充的比特个数是 $16-R[3:0]$ (16 比特并行处理) 或者 $32-R[4:0]$ (32 比特并行处理) 个 0, 按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特 (16 比特并行处理) 或者 128 比特 (32 比特并行处理) 的校验比特 1 和校验比特 2 数据。

10 按照列来进行解交织和解速率匹配, 当解交织的列数行数和 ncb_col, ncb_row 相同 (到译码卷绕的位置), 如果当前译码的长度还未达到 e 值, 分成以下 3 情况进行卷绕:

第一种:

15 $k0_flag=0$, $ncb_flag=0$, $process_flag$ 为 0, 判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 如果相同卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

第二种:

$k0_flag=0$, $ncb_flag=1$, $process_flag$ 为 0, 不判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

20 第三种:

$k0_flag=1$, $ncb_flag=1$, $process_flag$ 为 1, 判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

当解交织的数据长度达到 e 值, 数据解交织过程完成。

最后, 从缓存中读出待合并的解交织解速率匹配的数据。把解交织
25 的数据和速率匹配中存储的数据进行饱和处理, 处理后的数据和延迟单

元 210（延迟单元的功能就是将读地址延迟固定的 4 个时钟周期）的地址一起把数据写入 CB 解交织解速率匹配缓存之中。

可见，通过采用上述方案，就能采用高并行处理的解交织解速率匹配的方法，可以成倍的提高处理的能力，减少处理延迟。

5 另外，本申请的方法可以直接应用与 LTE 的基站和 UE 的设计之中，并且参考本申请并行度还可以提升到 64 或者 128，最高的并行度可以达到 193 比特，进一步减少解交织解速率匹配的处理延迟。

本实施例提供了一种解交织解速率匹配的装置，其实现原理图如图
10 1-2 所示，组成如下：

数据缓存单元 11，用于缓存至少一个待译码数据；

解交织解速率匹配控制单元 12，用于从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

15 数据解交织单元 13，用于对所述目标待译码数据进行解交织处理，得到解交织后的数据

速率匹配单元 14，用于对所述目标待译码数据进行解速率匹配处理，得到解速率匹配之后的数据；

解交织解速率匹配数据缓存单元 15，用于缓存解速率匹配以及解交织后的数据；

20 合并单元 16，用于将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

具体来说，参见图 1-3，其中：

第一部分：CB 译码数据缓存 102，主要的功能是缓存某个 CB 的解交织和解速率匹配前的数据，可以通过解交织解速率匹配控制模块提前
25 从外部的 TB 数据缓存区中搬移到 CB 的数据缓存区之中。

第二部分：CB 解交织解速率匹配控制 103，主要的功能是接收外部

配置的 TB 的解交织解速率匹配的任务，拆分成 CB 解交织解速率匹配的任务，产生读译码数据的地址，同时产生解交织模块的读写地址，控制解交织解速率匹配的过程。

第三部分：速率匹配读数据 104，主要的功能是根据控制模块 103 从速率匹配缓存区中读出对应的解交织的数据，缓存到内部的寄存器之中。

第四部分：数据解交织 105，主要的功能是根据控制模块提供的系统比特，校验 1 比特和校验 2 比特等控制信息，把系统比特，校验 1 比特和校验 2 比特分分离，分别单独的缓存到对应的寄存器之中。

第五部分：速率匹配 106，主要的功能是把从速率匹配的数据和解交织后的数据进行饱和累加，不管速率匹配是打孔的模式还是重复的模式，都是同样一个速率匹配的过程。

第六部分：CB 解交织解速率匹配数据缓存 107，主要的功能是用来缓存解交织解速率匹配后的数据，在解交织解速率匹配完成以后，该缓存区之中的数据就是解交织解速率匹配译码完成的数据，可以直接从该缓存区中读出给后面的模块 HARQ 合并 108 做后续的数据信道的处理。

第七部分：TB 数据缓存 101，主要是用来缓存多个 CB 的待译码的数据，一般选择缓存到比较大的内存或者 DDR 之中。

第八部分：HARQ 合并 108，主要是从解交织解速率匹配 107 中读取译码后的数据和上次 HARQ 重传的数据合并，合并后的数据在送给后续的 Turbo 译码器进行译码。

上述技术方案适用于 LTE 制式满足 3GPP 36212 协议规定的子块交织和速率匹配的逆过程，适用于 LTE 基站和 UE 侧的数据信道的解交织解速率匹配。

本申请提供了一种解交织解速率匹配的装置，其详细实现原理图如

图 2 所示，组成如下：

第一部分：TB 数据缓存 201，也就是整个 TB 的数据缓存区，一般选择存储容量大的 L2D 或者 DDR。

第二部分：CB 数据缓存区 202，可以采用 FIFO 或者 RAM 实现，
 5 LTE 的 $cbsize$ 大小从 40 到 6144，经过编码后速率匹配的长度是 e ，FIFO 的缓存区的大小可以设置为 e 个 byte，一般在译码的时候用 1 个 byte 来表示软信息大小，控制模块在 FIFO 有空间的时候，就把外部的 TB 数据缓存 201 读入到 CB 数据缓存区 202。

第三部分：CB 解交织解速率匹配控制 203，接收外部配置的 TB 任务，包括关键的参数 $k0$ ， ncb ， $cbsize$ ， e ， $k0$ 在速率匹配的过程之中是数据编码的开始位置， ncb 是速率匹配过程之中数据编码的卷绕位置，如果到了 ncb 卷绕位置，数据长度还未达到 e ，则编码重新从 0 开始取数据。
 10 $Cbsize$ 是 LTE 编码的 CB 的码块长度，是固定的 188 种长度，范围是 40 到 6144， e 是代表数据译码的长度，一般单位是 byte。

15 然后按照下面步骤计算出解交织解速率匹配需要的参数和解交织的 RAM 初始化：

第一步：计算交织矩阵的行数 R ， $R = cbsize / 32 + 1$ ，

第二步：计算出交织矩阵的填充比特个数 nd ， $nd = 28 - cbsize[4:0]$ 。

第三步：计算 $k0$ 的起始位置在矩阵是属于系统比特还是校验比特
 20 $k0_flag$ ，如果 $k0 < 32 * R$ ， $k0_flag = 0$ ，否则 $k0_flag = 1$ 。

第四步：计算 $k0$ 的起始位置在矩阵的列号 $k0_col$ ，如果 $k0 < 32 * R$ ，则
 $k0_col = \lfloor k0 / R \rfloor$ ，否则 $k0_col = \lfloor (k0 - 32 * R) / 2 / R \rfloor$ 。

第五步：计算 $k0$ 的起始位置在矩阵的列号 $k0_row$ ，如果 $k0 < 32 * R$ ，
 则 $k0_row = k0 - \lfloor k0 / R \rfloor * R$ ，否则 $k0_row = (k0 - 32 * R) / 2 - k0_col * R$ 。

25 第六步： ncb 的结束位置在矩阵是属于系统比特还是校验比特

ncb_flag ，如果 $ncb < 32 * R$ ， $ncb_flag = 0$ ，否则 $ncb_flag = 1$ 。

第七步：计算 ncb 的结束位置在矩阵的列号 ncb_col ，如果 $ncb < 32 * R$ ，则 $ncb_col = \lfloor ncb / R \rfloor$ ，否则 $ncb_col = \lfloor (ncb - 32 * R) / 2 / R \rfloor$ 。

第八步：计算 ncb 的结束位置在矩阵的行号 ncb_row ，如果 $ncb < 32 * R$ ，
5 则 $ncb_row = ncb - \lfloor ncb / R \rfloor * R$ ，否则 $ncb_row = (ncb - 32 * R) / 2 - ncb_col * R$ 。

第九步：CB 解交织解速率匹配控制 203 会发出 CB 解交织解速率匹配缓存 212 的所有写地址进行写操作，把该缓存区都写成 0，这样后面速率匹配的过程不管是打孔模式的速率匹配，还是重复模式的速率匹配，还是多个超级小区合并的解速率匹配，都统一到一种硬件处理单元之中。

10 在解速率的过程之中，设置 row_cnt 和 col_cnt 2 个变量，初始值是 $k0_row$ 和 $k0_col$ ，还设置 1 个 $process_flag$ ，初始值就是 $k0_flag$ 。

第四部分：读地址产生 204，该模块的读地址是按照顺序产生，从 CB 缓存区中顺序读取数据，下面都以 $cbsize$ 为 6144 的码块长度来说明解交织解速率匹配的长度， $cbsize$ 为 6144 的交织矩阵的行数是 193 行，读
15 地址产生模块每个 cycle 从 CB 缓存区 202 中读取 128 比特数据或者 256 比特数据，也就是 16 或者 32 个待解交织的软符号。

第五部分：译码数据 205，主要是用寄存器来缓存从 CB 数据缓存区读出的 128 比特或者 256 比特数据。

第六部分：速率匹配写地址 206，解交织解速率匹配的过程是按列
20 来解交织解速率匹配的，由于编码过程之中进行了列交换，在译码过程之中也要进行相应的列交换，交换的方法如下表 2 所示：

列	0	1	2	3	4	5	6	7
实际列	0	16	8	24	4	20	12	28
列	8	9	10	11	12	13	14	15

实际列	2	18	10	26	6	22	14	30
列	16	17	18	19	20	21	22	23
实际列	1	17	9	25	5	21	13	29
列	24	25	26	27	28	29	30	31
实际列	3	19	11	27	7	23	15	31

表 2

解速率匹配的过程是按照列来进行，在某列速率匹配的过程之中， col_cnt 保持不变，而 row_cnt 在系统比特解交织解速率匹配的系统比特的时候固定增加 16 或者 32；而在校验比特的时候， row_cnt 固定增加 8 或者 16。

写地址产生的规则就是 $col_cnt_write[0]+row_cnt[7:4]$ （16 比特并行处理）或者 $col_cnt_write[0]+row_cnt[7:5]$ （32 比特并行处理）。

第七部分：数据解交织 207，对数据进行解交织处理，分成 2 种情况

10 第一种： $process_flag$ 为 0，代表当前处理的数据是系统比特，解交织和解速率匹配的过程主要有 3 步骤：

15 第一步： $row_cnt=0$ 的时候，也就是在进行第零行的处理时，首先判断 col_cnt_write 是否小于 nd （交织矩阵第 0 行需要填充的无效比特个数），如果 $col_cnt_write < nd$ ，要那么在数据前面填充 1 个 byte 的 0，否则不用填充 1 个 byte 的 0；然后从填充后的数据中读取 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理）。

第二步： row_cnt 不是最后一行的情况，将读取的数据 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理）直接做为解交织后的数据。

20 第三步： row_cnt 是最后一行的时候，真实有效的数据不够 128 比特

或者 256 比特，需要填充比特。填充的比特个数是 $16-R[3:0]$ （16 比特并行处理）或者 $32-R[4:0]$ （32 比特并行处理）个 0，然后把填充后的数据作为真实的写数据。

第二种： *process_flag* 为 1，代表当前处理的数据是校验比特，解交织和解速率匹配的过程主要有 3 步骤：

第一步： *row_cnt*=0 的时候，判断 *col_cnt_write* 是否大于 *nd*，如果 *col_cnt_write*<*nd*，要在数据前面填充 2 个 byte 的 0，否则不用填充 2 个 byte 的 0。然后从填充后的数据中读取 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理），按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特（16 比特并行处理）或者 128 比特（32 比特并行处理）的校验比特 1 和校验比特 2 数据。

第二步： *row_cnt* 不是最后一行的情况，将读取的数据 128 比特（16 比特并行处理）或者 256 比特（32 比特并行处理），按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特（16 比特并行处理）或者 128 比特（32 比特并行处理）的校验比特 1 和校验比特 2 数据。

第三步： *row_cnt* 是最后一行的时候，真实有效的数据不够 128 比特或者 256 比特，需要填充比特。填充的比特个数是 $16-R[3:0]$ （16 比特并行处理）或者 $32-R[4:0]$ （32 比特并行处理）个 0，按照奇偶 byte 分别属于校验比特 1 校验比特 2 的方法分别产生 64 比特（16 比特并行处理）或者 128 比特（32 比特并行处理）的校验比特 1 和校验比特 2 数据。

CB 解交织解速率匹配控制模块控制整个装置，按照列来进行解交织和解速率匹配，当解交织的列数行数和 *ncb_col*, *ncb_row* 相同（到译码卷绕的位置），如果当前译码的长度还未达到 *e* 值，分成以下 3 情况进行卷绕：

第一种：

$k0_flag=0$, $ncb_flag=0$, $process_flag$ 为 0, 判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 如果相同卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

第二种:

5 $k0_flag=0$, $ncb_flag=1$, $process_flag$ 为 0, 不判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

第三种:

10 $k0_flag=1$, $ncb_flag=1$, $process_flag$ 为 1, 判断 row_cnt 和 col_cnt 是否和 ncb_col, ncb_row 相同, 卷绕的行列号是 $row_cnt=0$, $col_cnt=0$ 。

当解交织的数据长度达到 e 值, 数据解交织过程完成。

第八部分: 读地址 211, 把写地址 206 作为读地址, 从 CB 块解交织解速率匹配的缓存中读出待合并的解交织解速率匹配的数据。

15 第九部分: 速率匹配数据 208, 从 CB 解交织解速率匹配的缓存中选出和解交织后对应的待合并的数据。

第十部分: 速率匹配 209, 把解交织的数据和速率匹配中存储的数据进行饱和处理, 处理后的数据和延迟单元 210 (延迟单元的功能就是将读地址延迟固定的 4 个时钟周期) 的地址一起把数据写入 CB 解交织解速率匹配缓存之中。

20 上述过程具体描述一种 LTE 的数据信道的 16 比特或者 32 比特并行处理的解交织解速率匹配的过程, 该方法和装置实现简单, 解交织解速率匹配的过程把解打孔和解重复, 解交织, LTE 的超级小区合并都简单的统一到一个硬件流程之中, 硬件的译码的处理延迟是 $e/16$ (16 比特并行处理) 或者 $e/32$ (32 比特并行处理) 个 cycle, 成倍的节省了处理的延
25 迟。

上述的解交织解速率匹配的方法和装置同时可以实现 LTE 基站侧的数据信道的译码，也可以实现 LTE UE 侧的数据信道的译码，可以节省硬件的实现面积和功耗。

5 采用高并行处理的解交织解速率匹配的方法，可以成倍的提高处理的能力，减少处理延迟。本申请的方法可以直接应用与 LTE 的基站和 UE 的设计之中，并且参考本申请并行度还可以提升到 64 或者 128，最高的并行度可以达到 193 比特，进一步减少解交织解速率匹配的处理延迟。

10 本申请实现了一种 LTE 高并行度的解交织解速率匹配，有效的提升了处理能力，降低了处理延迟，并且能满足 LTE 基站和 UE 的处理要求，参考本方案并行度还可以得到更大的提升，可以进一步降低处理延迟。

15 在前述实施例的基础上，本申请还提供了一种计算机存储介质，该计算机存储介质包括一组计算机可执行指令，所述计算机可执行指令配置为执行：

从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

对所述目标待译码数据进行解交织处理以及解速率匹配处理，得到解交织后的数据、以及解速率匹配之后的数据；

20 将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

所述计算机可执行指令还配置为执行：计算得到所述解交织处理以及解速率匹配处理所需要的参数。

所述计算机可执行指令还配置为执行：目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

25 针对提取到的每一列数据，根据所述列数据中每一个数据对应的行

数，以及每一个数据对应的类型，确定增加的数值；

基于每一个数据对应的增加的数值，对所述目标待译码数据进行解速率匹配。

所述计算机可执行指令还配置为执行：目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

从所述每一列数据中提取得到每一行数据，判断所述每一行数据为系统比特或校验比特，基于所述判断结果对所述数据进行解交织。

所述计算机可执行指令还配置为执行：当所述判断结果表征所述数据为系统比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置填充为 0；

若所述数据处于第 N 行，直接将读取到的数据作为解交织后的数据；其中，N 为大于等于 1 的整数。

所述计算机可执行指令还配置为执行：当所述判断结果表征所述数据为校验比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置之前的指定比特数填充为 0；

若所述数据处于第 N 行、且所述数据不处于最后行，将读取的数据 128 比特或者 256 比特，按照奇偶分别产生 64 比特或者 128 比特的校验比特数据；

若所述数据处于最后行，则确定所处行中对应的填充比特以及填充比特数量。

进一步地，本申请提供的一种解交织解速率匹配的装置，包括：处理器和用于存储能够在处理器上运行的计算机程序的存储器，

其中，所述处理器用于运行所述计算机程序时，执行：

从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

对所述目标待译码数据进行解交织处理以及解速率匹配处理，得到解交织后的数据、以及解速率匹配之后的数据；

将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

- 5 所述计算机可执行指令还配置为执行：计算得到所述解交织处理以及解速率匹配处理所需要的参数。

所述处理器用于运行所述计算机程序时，还执行：

目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

- 10 针对提取到的每一列数据，根据所述列数据中每一个数据对应的行数，以及每一个数据对应的类型，确定增加的数值；

基于每一个数据对应的增加的数值，对所述目标待译码数据进行解速率匹配。

所述处理器用于运行所述计算机程序时，还执行：

- 15 目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

从所述每一列数据中提取得到每一行数据，判断所述每一行数据为系统比特或校验比特，基于所述判断结果对所述数据进行解交织。

所述处理器用于运行所述计算机程序时，还执行：

- 20 当所述判断结果表征所述数据为系统比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置填充为 0；

若所述数据处于第 N 行，直接将读取到的数据作为解交织后的数据；其中，N 为大于等于 1 的整数。

- 25 所述处理器用于运行所述计算机程序时，还执行：

当所述判断结果表征所述数据为校验比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置之前的指定比特数填充为 0；

若所述数据处于第 N 行、且所述数据不处于最后行，将读取的数据 128 比特或者 256 比特，按照奇偶分别产生 64 比特或者 128 比特的校验比特数据；

若所述数据处于最后行，则确定所处行中对应的填充比特以及填充比特数量。

本申请所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时，也可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机、网络设备、或者网络设备等）执行本申请各个实施例所述方法的全部或部分。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（Read-Only Memory, ROM）、随机存取存储器（Random Access Memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。这样，本申请不限制于任何特定的硬件和软件结合。

以上所述，仅为本申请的较佳实施例而已，并非用于限定本申请的保护范围。

工业实用性

本申请提供了一种解交织解速率匹配的方法、装置及计算机存储介质，能够从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；对所述目标待译码数据进行解交织处理以及解速率匹配处理，得到解交织后的数据、以及解速率匹配之后的数据；将解交织后的数据以及

数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。从而实现采用高并行处理的解交织解速率匹配的方法，可以成倍的提高处理的能力，减少处理延迟。

权利要求书

1、一种解交织解速率匹配的方法，所述方法包括：

从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

对所述目标待译码数据进行解交织处理以及解速率匹配处理，得到

5 解交织后的数据、以及解速率匹配之后的数据；

将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

2、根据权利要求 1 所述的方法，其中，所述对所述目标待译码数据进行解交织处理以及解速率匹配处理之前，所述方法还包括：

10 计算得到所述解交织处理以及解速率匹配处理所需要的参数。

3、根据权利要求 2 所述的方法，其中，所述计算得到所述解交织处理以及解速率匹配处理所需要的参数，包括以下至少之一：

15 计算交织矩阵的行数 R ；计算出交织矩阵的填充比特个数 nd ；计算起始位置 $k0$ 所对应的类型标识 $k0_flag$ ；计算 $k0$ 在矩阵的列号 $k0_col$ ；计算 $k0$ 在矩阵的列号 $k0_row$ ；速率匹配的数据的结束位置 ncb 所对应的类型标识 ncb_flag ；计算 ncb 的在矩阵的列号 ncb_col ；计算 ncb 在矩阵的行号 ncb_row 。

4、根据权利要求 1 所述的方法，其中，所述对所述目标待译码数据进行解速率匹配处理，包括：

20 目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

针对提取到的每一列数据，根据所述列数据中每一个数据对应的行数，以及每一个数据对应的类型，确定增加的数值；

基于每一个数据对应的增加的数值，对所述目标待译码数据进行解

速率匹配。

5、根据权利要求 1 所述的方法，其中，所述对所述目标待译码数据进行解交织处理，包括：

目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

从所述每一列数据中提取得到每一行数据，判断所述每一行数据为系统比特或校验比特，基于所述判断结果对所述数据进行解交织。

6、根据权利要求 5 所述的方法，其中，所述基于所述判断结果对所述数据进行解交织，包括：

当所述判断结果表征所述数据为系统比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置填充为 0；

若所述数据处于第 N 行，直接将读取到的数据作为解交织后的数据；其中，N 为大于等于 1 的整数。

7、根据权利要求 5 所述的方法，其中，所述基于所述判断结果对所述数据进行解交织，包括：

当所述判断结果表征所述数据为校验比特时，若所述数据处于第零行，则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置之前的指定比特数填充为 0；

若所述数据处于第 N 行、且所述数据不处于最后行，将读取的数据 128 比特或者 256 比特，按照奇偶分别产生 64 比特或者 128 比特的校验比特数据；

若所述数据处于最后行，则确定所处行中对应的填充比特以及填充比特数量。

8、一种解交织解速率匹配的装置，所述装置包括：

数据缓存单元，用于缓存至少一个待译码数据；

解交织解速率匹配控制单元，用于从包括有至少一个待译码数据的缓存区中，读取目标待译码数据；

数据解交织单元，用于对所述目标待译码数据进行解交织处理，得到解交织后的数据；

速率匹配单元，用于对所述目标待译码数据进行解速率匹配处理，得到解速率匹配之后的数据；

解交织解速率匹配数据缓存单元，用于缓存解速率匹配以及解交织后的数据；

合并单元，用于将解交织后的数据以及数据匹配之后的数据进行合并，得到目标待译码数据对应的目标数据。

9、根据权利要求8所述的装置，其中，所述解交织解速率匹配控制单元，用于计算得到所述解交织处理以及解速率匹配处理所需要的参数。

10、根据权利要求9所述的装置，其中，所述解交织解速率匹配控制单元，用于执行以下计算至少之一：

计算交织矩阵的行数 R ；计算出交织矩阵的填充比特个数 nd ；计算起始位置 $k0$ 所对应的类型标识 $k0_flag$ ；计算 $k0$ 在矩阵的列号 $k0_col$ ；计算 $k0$ 在矩阵的列号 $k0_row$ ；速率匹配的数据的结束位置 ncb 所对应的类型标识 ncb_flag ；计算 ncb 的在矩阵的列号 ncb_col ；计算 ncb 在矩阵的行号 ncb_row 。

11、根据权利要求8所述的装置，其中，所述速率匹配单元，用于目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述目标待译码数据为矩阵数据；

针对提取到的每一列数据，根据所述列数据中每一个数据对应的行数，以及每一个数据对应的类型，确定增加的数值；

基于每一个数据对应的增加的数值，对所述目标待译码数据进行解速率匹配。

12、根据权利要求 8 所述的装置，其中，所述数据解交织单元，用于目标待译码数据的至少一列数据中分别提取每一列数据；其中，所述
5 目标待译码数据为矩阵数据；

从所述每一列数据中提取得到每一行数据，判断所述每一行数据为系统比特或校验比特，基于所述判断结果对所述数据进行解交织。

13、根据权利要求 12 所述的装置，其中，所述数据解交织单元，用于当所述判断结果表征所述数据为系统比特时，若所述数据处于第零行，
10 则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置填充为 0；

若所述数据处于第 N 行，直接将读取到的数据作为解交织后的数据；其中，N 为大于等于 1 的整数。

14、根据权利要求 12 所述的装置，其中，所述数据解交织单元，用于当所述判断结果表征所述数据为校验比特时，若所述数据处于第零行，
15 则对所述数据所处的列数是否小于预设数值，若小于，则控制将所述数据位置之前的指定比特数填充为 0；

若所述数据处于第 N 行、且所述数据不处于最后行，将读取的数据 128 比特或者 256 比特，按照奇偶分别产生 64 比特或者 128 比特的校验
20 比特数据；

若所述数据处于最后行，则确定所处行中对应的填充比特以及填充比特数量。

15、一种计算机存储介质，该计算机存储介质包括一组计算机可执行指令，所述计算机可执行指令配置为执行权利要求 1 至 7 任一项所述的方法的步骤。
25

16、一种解交织解速率匹配的装置，包括：处理器和用于存储能够在处理器上运行的计算机程序的存储器，

其中，所述处理器用于运行所述计算机程序时，执行权利要求 1 至 7 任一项所述方法的步骤。

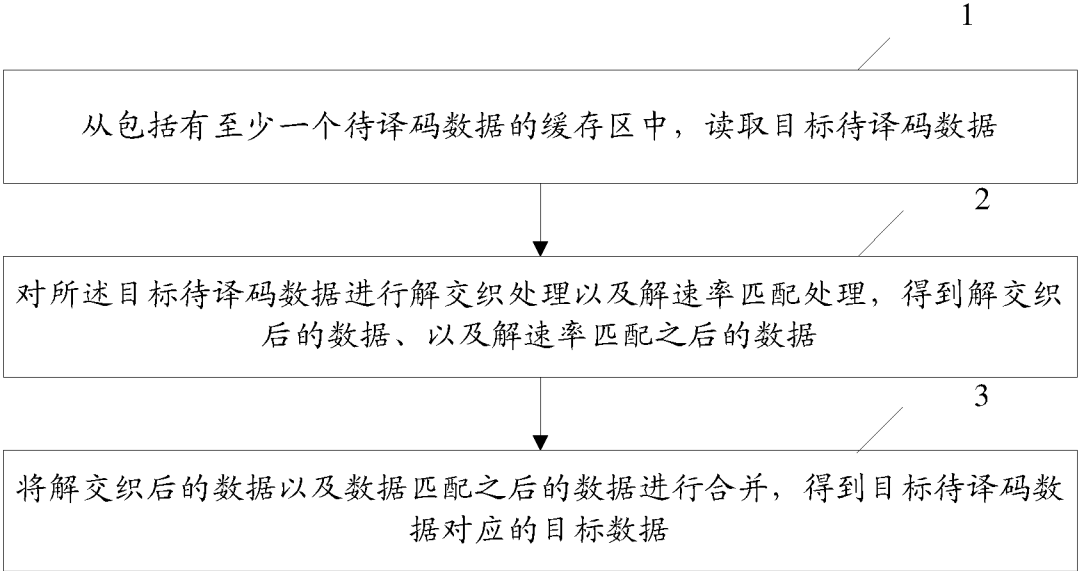


图 1-1

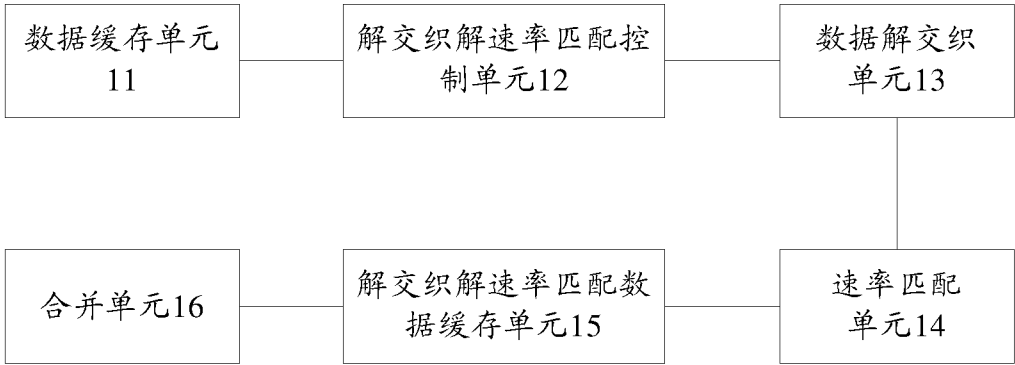


图 1-2

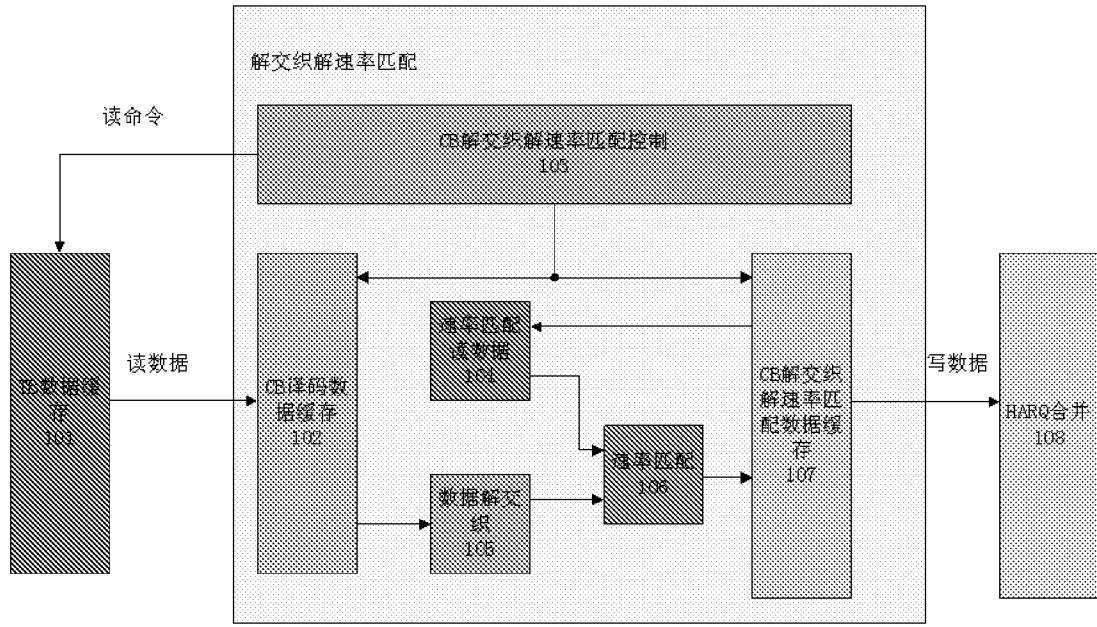


图 1-3

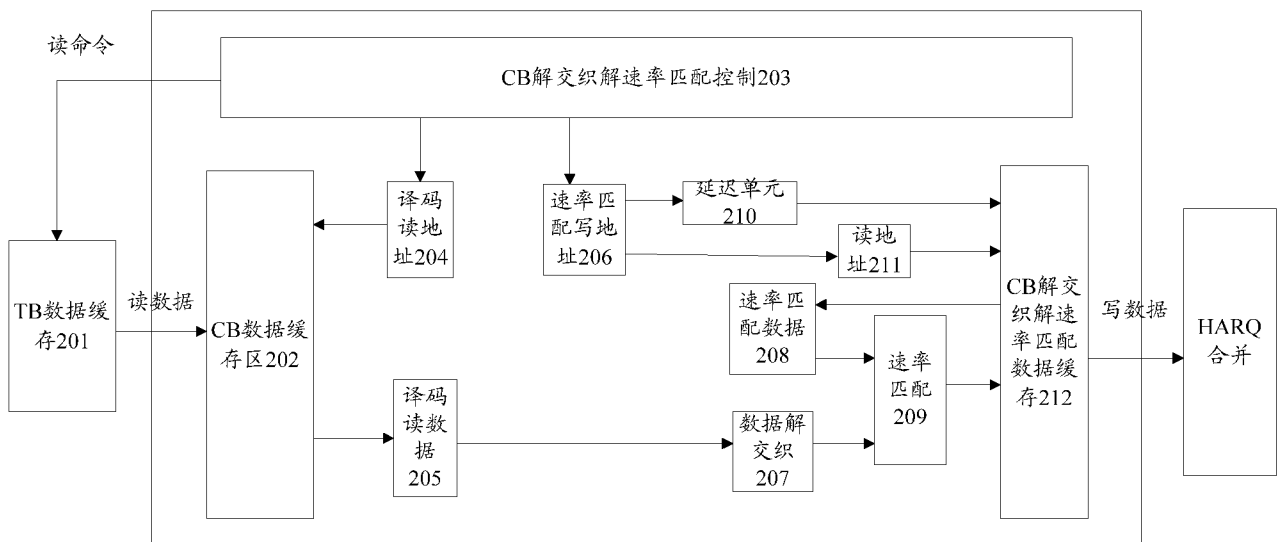


图 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/089808

A. CLASSIFICATION OF SUBJECT MATTER

H04L 1/00 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: 解交织, 解速率, 匹配, 缓存, 缓冲, 合并, 饱和, LTE, de?interleaving, rate, de? matching, buffer, cache, merge, combine, saturate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 101674159 A (ZTE CORP.), 17 March 2010 (17.03.2010), description, page 5, paragraph 8 to page 10, paragraph 21, and claims 1-11	1-16
Y	CN 101499875 A (SAMSUNG SDI CO., LTD. et al.), 05 August 2009 (05.08.2009), claims 1-15	1-16
A	CN 101087181 A (ZTE CORP.), 12 December 2007 (12.12.2007), entire document	1-16
A	CN 101090293 A (HUAWEI TECHNOLOGIES CO., LTD.), 19 December 2007 (19.12.2007), entire document	1-16
A	WO 2015028843 A1 (FREESCALE SEMICONDUCTOR, INC. et al.), 05 March 2015 (05.03.2015), entire document	1-16

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 15 August 2017	Date of mailing of the international search report 06 September 2017
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer WANG, Lunjie Telephone No. (86-10) 6164-8545

International application No.
PCT/CN2017/089808

Form PCT/ISA/210 (patent family annex) (July 2009)

国际检索报告

国际申请号

PCT/CN2017/089808

A. 主题的分类

H04L 1/00(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, CNKI, WPI, EPDOC:解交织, 解速率, 匹配, 缓存, 缓冲, 合并, 饱和, LTE, de?interleaving, rate, de? matching, buffer, cache, merge, combine, saturate

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
Y	CN 101674159 A (中兴通讯股份有限公司) 2010年 3月 17日 (2010 - 03 - 17) 说明书第5页第8段-第10页第21段, 权利要求1-11	1-16
Y	CN 101499875 A (三星电子株式会社等) 2009年 8月 5日 (2009 - 08 - 05) 权利要求1-15	1-16
A	CN 101087181 A (中兴通讯股份有限公司) 2007年 12月 12日 (2007 - 12 - 12) 全文	1-16
A	CN 101090293 A (华为技术有限公司) 2007年 12月 19日 (2007 - 12 - 19) 全文	1-16
A	WO 2015028843 A1 (FREESCALE SEMICONDUCTOR, INC. 等) 2015年 3月 5日 (2015 - 03 - 05) 全文	1-16

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2017年 8月 15日

国际检索报告邮寄日期

2017年 9月 6日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

王伦杰

电话号码 (86-10)6164-8545

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2017/089808

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	101674159	A	2010年 3月 17日	无			
CN	101499875	A	2009年 8月 5日	无			
CN	101087181	A	2007年 12月 12日	无			
CN	101090293	A	2007年 12月 19日	无			
WO	2015028843	A1	2015年 3月 5日	US	2016212644	A1	2016年 7月 21日

表 PCT/ISA/210 (同族专利附件) (2009年7月)