(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102064910 A
(43)申请公布日 2011.05.18

(21)申请号 200910224749.3

(22)申请日 2009.11.13

(71)申请人 傲世通科技(苏州)有限公司
地址 215021 江苏省苏州市苏州工业园区机场路328号国际科技园4期A1602室

(72)发明人 曾衡东 方明 朱志明 赖志强

(74)专利代理机构 南京苏科专利代理有限责任公司 32102
代理人 陈忠辉 姚姣阳

(51)Int.Cl.
H04L 1/00(2006.01)
H04L 1/18(2006.01)

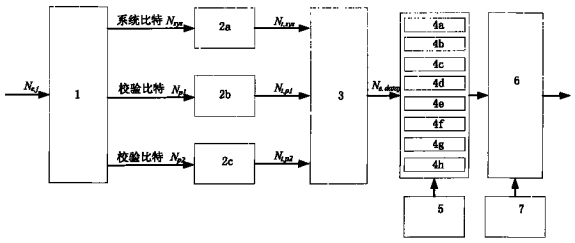
权利要求书 1 页 说明书 8 页 附图 6 页

(54)发明名称

混合自动重传请求比特收集和交织的方法和装置

(57)摘要

本发明涉及一种混合自动重传请求比特收集和交织的方法和装置，特点是将比特收集交织矩形定义为前后两个数据部分，将其中的数据比特分为八个序列，分别写入八个存储单元。随后，把八个存储单元当作一个整体的存储器处理，最后进行比特加扰和物理信道映射处理。应用本方法的装置，包括有比特分离模块，其输出端连接有三路速率匹配模块，该模块的输出端共同连入比特收集模块，比特收集模块的输出端连入八个储存单元，八个储存单元的控制端上连接有比特收集、输出控制单元，且储存单元的输出端共同连入后级比特加扰、物理信道映射处理单元的输入端。由此，不需要增加速率匹配模块与比特收集模块之间的缓存就能够实现混合自动重传请求比特收集和交织。



CN 102064910 A

1. 混合自动重传请求比特收集和交织的方法，其特征在于包括以下步骤：

步骤①，将比特收集交织矩形定义为前后两个数据部分，将其中的数据比特分为八个序列，分别写入八个存储单元；

步骤②，把八个存储单元当作一个整体的存储器处理，根据交织算法计算出当前所需数据在存储器的地址，从八个存储单元同时取出数据并还原数据流；

步骤③，进行比特加扰和物理信道映射处理。

2. 根据权利要求 1 所述的混合自动重传请求比特收集和交织的方法，其特征在于：步骤①所述的数据比特通过比特收集、输出控制单元按所定义的八个数据序列分别写入八个存储单元，所定义的比特收集交织矩形前后两部分数据中，同一部分的同一个序列数据，其写入存储单元的次序与对应的速率匹配模块输出结果的先后次序一致。

3. 根据权利要求 1 所述的混合自动重传请求比特收集和交织的方法，其特征在于：步骤③所述的比特加扰处理在比特收集的数据流成型之后，从八个存储单元同一个地址读取数据共获得 64 比特的数据，并行进行 64 位的比特加扰运算。

4. 混合自动重传请求比特收集和交织的装置，包括有比特分离模块，其特征在于：所述的比特分离模块的输出端连接有三路速率匹配模块的输入端，三路速率匹配模块的输出端共同连入比特收集模块的输入端，比特收集模块的输出端连入八个储存单元的输入端，八个储存单元的控制端上连接有比特收集、输出控制单元，八个储存单元的输出端共同连入后级比特加扰、物理信道映射处理单元的输入端，后级比特加扰、物理信道映射处理单元的控制端上连接后级交织器输入控制单元；所述八个存储单元用于存储比特收集的处理结果；所述比特收集、输出控制单元用于控制完成比特收集算法和输出到八块存储器的控制；所述后级交织器输入控制单元用于读出比特收集处理结果并重整数据还原数据流；所述后级比特加扰和交织处理单元用于 E-DCH 信道的比特加扰和交织地址运算处理。

5. 根据权利要求 4 所述的混合自动重传请求比特收集和交织的装置，其特征在于：所述的后级交织器输入控制单元采用并行处理。

混合自动重传请求比特收集和交织的方法和装置

技术领域

[0001] 本发明涉及一种第三代移动通信系统比特收集和交织的方法和装置，尤其涉及TD-SCDMA（时分同步码分多址）高速上行链路分组接入的一种混合自动重传请求比特收集和交织的方法和装置。

背景技术

[0002] 随着3GPP HSDPA(High Speed Downlink Packet Access，高速下行链路分组接入）标准化的完成，3G（第三代移动通信）系统对下行分组数据业务的支持能力有了很大的提高，为了满足用户对上行传输的性能需求，3GPP在HSDPA规范的基础上将自适应调制和编码、混合自动重传请求(HARQ)、高阶调制以及快速调度等关键技术应用于上行分组数据业务，从而形成HSUPA(High Speed Uplink Packet Access，高速上行链路分组接入）标准。HSUPA是一些无线增强技术的集合，利用HSUPA技术可以在现有技术的基础上使得上行峰值速率大大提高，并在上行链路得到更大的吞吐量。

[0003] 为了支持HSUPA特性，TD-SCDMA系统上行新增加了增强上行链路专用信道(E-DCH)，这是一个传输信道，用于承载高速上行数据。其传输时间间隔(TTI))为5ms，支持高阶调制，以及层1(L1)HARQ过程。其使用的资源，包括功率、时隙、码道等，可由NodeB调度分配。

[0004] 同时还定义了两个控制上行信道：上行增强控制信道(E-UCCH)和上行增强随机接入信道(E-RUCCH)，用于传输上行增强相关的信令信息。E-UCCH通常和E-DCH复用在一起，传递当前E-DCH HARQ相关的信息。E-RUCCH映射在物理随机接入资源上，主要用于上行增强业务的接入请求。

[0005] 在下行方向，为了支持基站调度，增加了增强上行绝对接入允许信道(E-AGCH)传输基站调度信息，以及增强上行HARQ应答指示信道(E-HICH)来支持HARQ过程的传输应答信息（如ACK/NACK)。

[0006] HSUPA的调度过程简述如下：

[0007] (1)UE（用户终端）通过E-RUCCH发起调度请求，调度请求包含调度相关信息以及UE的标识——无线网络临时标识(E-RNTI)。调度信息包括本小区和邻小区的路径损耗信息、可以允许使用的功率、缓存占用状况等等。

[0008] (2)NodeB调度器接收到请求后，若允许该UE发送上行增强数据，将通过E-AGCH发送接入允许信息给UE，接入允许信息主要包括功率允许和物理资源允许。并且由于E-AGCH是共享信道，因此接入允许信息还需要携用户标识区分该接入允许是给哪个UE的，同时还指示UE，其接收应答信息的E-HICH信道标识。

[0009] (3)UE收到E-AGCH，解得信息是给自己的后，就根据分配的资源和功率在E-DCH上选择自己可以使用的速率并开始数据传输，具有接入允许的UE，可以在MAC-e头重新携带调度信息。

[0010] (4)NodeB接收E-DCH信息，解调后根据数据是否正确，在该用户监听的E-HICH信

道上反馈 ACK/NACK 信息。UE 根据反馈信息判断是否需要重传。

[0011] HSUPA 系统中，终端收到 E-AGCH 之后，将在指定的 E-DCH 许可资源 E-PUCH 上发送数据，在终端侧，大量的数据是缓存在 RLC 的缓存中。在短时间内，终端将解读 E-AGCH，并且根据 E-AGCH 的内容进行 E-TFC 的选择，然后将 RLC 数据下传到 MAC 层，并且组装成 MAC-e PDU 数据包。而在 HSDPA 中是不存在这些过程，所以在 HSUPA 中，对芯片的处理能力提出了更高的要求。

[0012] 图 1 所示给出了 E-DCH 的信道编码链路图。从图中可以看出，E-DCH 除了增加 HARQ、和 16QAM 星座重排过程，与普通的信道编码过程并无太大区别。

[0013] 图 2 所示给出了 E-DCH HARQ 的功能框图。Turbo 编码器出来的数据 (N_{ej}) 经过比特分离算法分为系统比特 (N_{sys})，校验比特 1(N_{p1}) 和校验比特 2(N_{p2}) 三路数据流，分别将三路数据流送入各自的速率匹配器进行速率匹配运算。速率匹配后的结果 ($N_{t,sys}$，$N_{t,p1}$ 和 $N_{t,p2}$) 通过比特收集器合并成为一路数据流 ($N_{e,dataj}$) 送入后级比特加扰器和交织器。

[0014] TD-HSUPA 系统中 E-DCH 的 HARQ 比特收集与 HSDPA 系统 HS-DSCH 相同。采用 $N_{row} \times N_{col}$ 的矩形交织器。

[0015] 对于调制方式为 16QAM 的 E-DCH，N_{row} 等于 4，对于调制方式为 QPSK 的 E-DCH，N_{row} 等于 2。同时定义 N_{col} 为：

[0016] $N_{col} = N_{data}/N_{row}$

[0017] 这里 N_{data} 是一个 TTI 中 E-DCH 的有效的比特数。

[0018] 数据按列写入交织器，读出时从第一列开始一列接一列地读出来。

[0019] $N_{t,sys}$ 定义为传输系统比特的个数．中间量 N_r 和 N_c 定义如下：：

[0020] $$N_r = \left\lfloor \frac{N_{t,sys}}{N_{col}} \right\rfloor \text{ and } N_c = N_{t,sys} - N_r \cdot N_{col}.$$

[0021] 如果 $N_c = 0$ 并且 $N_r > 0$，则系统比特写入交织矩形的 1 到 N_r 行，否则，系统比特在前 N_c 列写入 1 到 N_r+1 行，如果此时 $N_r > 0$，那么系统比特在剩下的 $N_{col}-N_c$ 列写入 1 到 N_r 行。

[0022] 系统比特填充完成后剩余的空间用于填充校验比特。校验比特也按列写入剩余行的相应列中。校验比特 1 和校验比特 2 交替写入。从校验比特 2 开始，序号最低的写入第一个相应的列中。

[0023] 数据从比特收集器第一列到最后一列依次输出，每一列按照以下方式操作：

[0024] 对于 16QAM 调制方式，交织矩形的每一列数据按照行 1，行 2，行 3，行 4 的顺序输出，对于 QPSK 调制方式，交织矩形的每一列数据按照行 1，行 2 的顺序输出。

[0025] 根据上述 3GPP 规范的定义，图 3 给出了系统比特，校验比特 1，校验比特 2 在比特收集交织矩形中所有可能的排列顺序。图 3(a) 为 16QAM 调制方式下没有校验比特，全部为系统比特的情况。图 3(b) 为 16QAM 调制方式下系统比特数大于三行小于四行的情况。图 3(c) 为 16QAM 调制方式下系统比特数正好三行的情况。图 3(d) 为 16QAM 调制方式下系统比特数大于两行小于三行的情况，并且 N_c 为奇数。图 3(e) 为 16QAM 调制方式下系统比特数大于两行小于三行的情况，并且 N_c 为偶数。图 3(f) 为 16QAM 调制方式下系统比特数正好两行的情况。图 3(g) 为 16QAM 调制方式下系统比特数大于一行小于两行的情况。图 3(h) 为 16QAM 调制方式下系统比特数正好一行的情况。图 3(i) 为 16QAM 调制方式下系统比特

数不足一行的情况，并且N_c为奇数。图3(j)为16QAM调制方式下系统比特数不足一行的情况，并且N_c为偶数。图3(k)为16QAM调制方式下系统比特数为零的情况。图3(l)为QPSK调制方式下没有校验比特，全部为系统比特的情况。图3(m)为QPSK调制方式下系统比特数大于一行小于两行的情况。图3(n)为QPSK调制方式下系统比特数正好一行的情况。图3(o)为QPSK调制方式下系统比特数不足一行的情况，并且N_c为奇数。图3(p)为QPSK调制方式下系统比特数不足一行的情况，并且N_c为偶数。图3(q)为QPSK调制方式下系统比特为零的情况。

[0026] 从图3中可以看出，比特收集交织矩形排列的情况比较多，而且turbo编码器输出的结果通过比特分离分为系统比特，校验比特1和校验比特2三路数据流，并各自进行速率匹配运算，速率匹配的结果导致三路数据流先后次序被打乱，因而给比特收集的处理带来麻烦。

[0027] 图4(a)，图4(b)和图5给出了现有技术的HARQ比特收集和交织的方法。图4(a)和图4(b)区别在于交织的实现方式不同，图4(a)是将比特收集后的数据先做交织，然后将交织后的数据保存到存储单元以备后续处理。图4(b)是将比特收集后的数据先存储起来，然后再做后续运算的时候通过交织算法找到当前需要的数据，两种方法没有本质的区别。从图4和图5中可以看出现有技术存在如下的缺点：

[0028] 1、图4(a)，图4(b)都采用了一种传统的方法，原理是将速率匹配后的三路乱序的数据各自缓冲起来，然后再通过控制逻辑从缓冲器中取出相应的数据做比特收集。但TD-HSUPA的数据量较大，系统比特最大可能为14080bits，校验比特1和校验比特2数目最大可能为7040bits。采用图4的方法增加了3块较大的存储资源的开销。

[0029] 2、因为比特收集交织矩形的排列情况很多，即使通过数据缓存后再做比特收集，也需要复杂的控制电路来实现，势必增加逻辑开销，造成资源浪费，而且缓存的方法需要分两级来完成，降低了电路的工作效率。

[0030] 3、图5的方法通过3个位宽为1的FIFO来缓存比特收集前的速率匹配的结果，FIFO的深度根据系统仿真的最大流量来决定，当FIFO满时通过控制逻辑禁止前级Turbo编码数据流写入。这种方法虽然不需要将所有速率匹配后的数据都缓存起来，比图4的方法减少了部分存储资源，但3个位宽为1的FIFO和控制逻辑也存在不小的硬件开销，同时FIFO的状态还影响到turbo编码器的工作，这也增加了执行时间上的开销。

[0031] 4、规范定义的都是比特位的操作，现有技术通过位寻址的缓存方式不能有效地节约硬件资源。

[0032] 因此，现有技术存在很多不足，需要进一步发展和改进。

发明内容

[0033] 本发明的目的就是为了解决现有技术中存在的上述问题，提供一种混合自动重传请求比特收集和交织的方法和装置。

[0034] 本发明的目的通过以下技术方案来实现：

[0035] 混合自动重传请求比特收集和交织的方法，其包括以下步骤：

[0036] 步骤①，将比特收集交织矩形定义为前后两个数据部分，将其中的数据比特分为八个序列，分别写入八个存储单元；

[0037] 步骤②，把八个存储单元当作一个整体的存储器处理，根据交织算法计算出当前所需数据在存储器的地址，从八个存储单元同时取出数据并还原数据流；

[0038] 步骤③，进行比特加扰和物理信道映射处理。

[0039] 上述的混合自动重传请求比特收集和交织的方法，其中：步骤①所述的数据比特通过比特收集、输出控制单元按所定义的八个数据序列分别写入八个存储单元，所定义的比特收集交织矩形前后两部分数据中，同一部分的同一个序列数据，其写入存储单元的次序与对应的速率匹配模块输出结果的先后次序一致。

[0040] 进一步地，上述的混合自动重传请求比特收集和交织的方法，其中：步骤③所述的比特加扰处理在比特收集的数据流成型之后，从八个存储单元同一个地址读取数据共获得64 比特的数据，并行进行 64 位的比特加扰运算。

[0041] 混合自动重传请求比特收集和交织的装置，包括有比特分离模块，其特征在于：所述的比特分离模块的输出端连接有三路速率匹配模块的输入端，三路速率匹配模块的输出端共同连入比特收集模块的输入端，比特收集模块的输出端连入八个储存单元的输入端，八个储存单元的控制端上连接有比特收集、输出控制单元，八个储存单元的输出端共同连入后级比特加扰、物理信道映射处理单元的输入端，后级比特加扰、物理信道映射处理单元的控制端上连接后级交织器输入控制单元；所述八个存储单元用于存储比特收集的处理结果；所述比特收集、输出控制单元用于控制完成比特收集算法和输出到八块存储器的控制；所述后级交织器输入控制单元用于读出比特收集处理结果并重整数据还原数据流；所述后级比特加扰和交织处理单元用于 E-DCH 信道的比特加扰和交织地址运算处理。

[0042] 上述的混合自动重传请求比特收集和交织的装置，其中：所述的后级交织器输入控制单元采用并行处理。

[0043] 本发明技术方案的突出的实质性特点和显著的进步主要体现在：不需要增加缓存就能够实现混合自动重传请求比特收集和交织，通过改变现有技术的交织所用的存储器的结构，在比特收集结果存储和后级交织输入适当增加控制逻辑，完全避免了速率匹配到比特收集间缓存的使用，减小了电路面积，提高了硬件的工作效率。同时新的比特收集结果的存储结构也方便了后级两路交织处理的并行实现，进一步减小了整个物理过程的延迟时间，提高了 TD-HSUPA 系统的处理能力。

附图说明

[0044] 本发明的目的、优点和特点，将通过下面优选实施例的非限制性说明进行图示和解释。这些实施例仅是应用本发明技术方案的典型范例，凡采取等同替换或者等效变换而形成的技术方案，均落在本发明要求保护的范围之内。这些附图当中，

[0045] 图 1 是 3GPP 规范定义的增强上行链路专用信道 (E-DCH) 的编码链路图；

[0046] 图 2 是 3GPP 规范定义的增强上行链路专用信道 (E-DCH) 的 HARQ 功能框图；

[0047] 图 3(a) ～ 3(q) 是 3GPP 规范定义的比特收集矩形交织器的数据排列示意图；

[0048] 图 4(a)、3(b) 是现有技术中采用缓存的 HARQ 比特收集和交织的功能框图；

[0049] 图 5 是现有技术中采用 FIFO 缓冲的 HARQ 比特收集和交织的功能框图；

[0050] 图 6 是混合自动重传请求比特收集和交织装置的 HARQ 比特收集和交织的功能框图；

[0051] 图 7 是采用混合自动重传请求比特收集和交织的方法后在 16QAM 调制方式下比特收集矩形交织器示意图；

[0052] 图 8(a) ～ 8(h) 是与图 7 对应的存储单元示意图；

[0053] 图 9(a)、9(b) 是采用混合自动重传请求比特收集和交织的方法后，比特收集矩形交织器划分为两个数据部分的示意图；

[0054] 图 10 是 3GPP 规范规定的 E-DCH 交织功能结构框图。

[0055] 图中各附图标记的含义如下：

[0056] 1 比特分离模块 2a ～ 2c 速率匹配模块

[0057] 3 比特收集模块 4a ～ 4h 储存单元

[0058] 比特加扰、物理信道映

[0059] 5 比特收集、输出控制单元 6

[0060] 射处理单元

[0061] 7 交织器输入控制单元

具体实施方式

[0062] 混合自动重传请求比特收集和交织的方法，其特别之处在于包括以下步骤：首先，将比特收集交织矩形定义为前后两个数据部分，将其中的数据比特分为八个序列，分别写入八个存储单元。随后，把八个存储单元当作一个整体的存储器处理，根据交织算法计算出当前所需数据在存储器的地址，从八个存储单元同时取出数据并还原数据流；最后，进行比特加扰和物理信道映射处理。

[0063] 结合本发明一较佳的实施方式来看，数据比特通过比特收集、输出控制单元按所定义的八个数据序列分别写入八个存储单元，所定义的比特收集交织矩形前后两部分数据中同一部分的同一个序列数据写入存储单元的次序与对应的速率匹配模块输出结果的先后次序一致。并且，比特加扰处理在比特收集的数据流成型之后，从八个存储单元同一个地址读取数据共获得 64 比特的数据，并行进行 64 位的比特加扰运算。

[0064] 如图 6 所示的混合自动重传请求比特收集和交织的装置，包括有比特分离模块 1，其与众不同之处在于：所述的比特分离模块 1 的输出端连接有三路速率匹配模块 2a ～ 2c 的输入端，三路速率匹配模块 2a ～ 2c 的输出端共同连入比特收集模块 3 的输入端，比特收集模块 3 的输出端连入八个储存单元 4a ～ 4h 的输入端，八个储存单元 4a ～ 4h 的控制端上连接有比特收集、输出控制单元 5，八个储存单元 4a ～ 4h 的输出端共同连入后级比特加扰、物理信道映射处理单元 6 的输入端，后级比特加扰、物理信道映射处理单元 6 的控制端上连接后级交织器输入控制单元 7。

[0065] 进一步来看，所述八个存储单元用于存储比特收集的处理结果。所述比特收集、输出控制单元 5 用于控制完成比特收集算法和输出到八块存储器的控制。后级交织器输入控制单元 7 用于读出比特收集处理结果并重整数据还原数据流。后级比特加扰和交织处理单元用于 E-DCH 信道的比特加扰和交织地址运算处理。并且，为了便于输入，后级交织器输入控制单元 7 采用并行处理。

[0066] 以 16QAM 调制方式为例即图 7 的比特收集矩形交织器示意图，将本发明的方法和装置的具体实施进行较为详细的说明：

[0067] 如图 7、8(a) ～ (h) 所示根据规范规定，比特收集的矩形交织器分为四行，本发明将奇数列的第一行作为第一个序列，将奇数列的第二行作为第二序列，将奇数列的第三行作为第三序列，将奇数列的第四行作为第四序列，将偶数列的第一行作为第五序列，将偶数列的第二行作为第六序列，将偶数列的第三行作为第七序列，将偶数列的第四行作为第八序列。根据序号将相关的数据序列写入相同序号的存储单元，总共八个存储单元。

[0068] 为了方便描述，将比特收集矩形交织器中的数据比特用 $m_{x,y}$ 表示，其中 x 表示本发明定义的八个数据序列的序列号，x 的取值范围从 0 到 7，分别对应八个数据序列，同时也对应八个存储单元。y 表示每一个序列中数据的序号，同一列中的比特的 y 序号相同。

[0069] 如图 9(a) 和图 9(b) 所示，以 N_c 为分界线将比特收集的矩形交织器分为两个部分，前 N_c 列为第一部分，$N_{col}-N_c$ 列为第二部分 (N_c 为 0 的情况只有第二部分)，以系统比特不足一行的两种情况为例。根据如图 3 所示的所有可能的比特收集矩形交织器数据排列的情况分析，在第一部分数据块中排列在同一行中奇数列的数据来自速率匹配后的同一个数据源 (系统比特，校验比特 1 或校验比特 2)，而且该数据在数据源中的排列顺序按列号增加而增加。同样，在第一部分数据块中排列在同一行中偶数列的数据也来自同一个数据源，数据在数据源中的排列顺序按列号增加而增加。用 $m_{x,y}$ 表示数据比特，其规律可以解释为，在第一部分数据中，x 序号相同的数据来自同一个数据源，y 序号随列号增加而增加，并且 y 的大小次序与速率匹配输出的该数据源的比特先后次序相同，y 值小的比特先输出。在第二部分数据中也存在同样的规律。

[0070] 根据 N_{col}，N_c 和 N_r 的值可以计算出八个数据序列 (即八个存储单元) 的数据源头，而每个数据序列的比特排列顺序是和其在速率匹配输出的先后次序对应的。因此将速率匹配后输出到各个数据序列的数据进行拼接后写入其对应的存储单元成为可能，避免了采用位寻址而增加的存储器结构和面积的资源开销。本示例中采用位宽为 8 的存储器，如图 7 所示。当做后级交织处理时，通过交织算法计算出所需数据在存储单元的地址，同时从八块存储器同一地址取出数据，八块存储器总共获得 64 比特的数据，只需对这 64 比特数据做一个次序重排即得到需要的比特收集的数据流。

[0071] 还原后的数据流并行进行比特加扰处理，并且做后级的物理信道映射等处理。

[0072] 具体的控制流程可以分为以下几个步骤 :

[0073] 1、经过比特分离后的系统比特，校验比特 1 和校验比特 2 并行进入各自的速率匹配器做速率匹配运算。

[0074] 2、根据 N_c 将比特收集的交织矩形分为如图 9(a)、9(b) 所示的前 N_c 列和后 $N_{col}-N_c$ 列两部分数据块。

[0075] 3、根据比特收集参数 (N_{col}，N_c 和 N_r) 分别将系统比特，校验比特 1 或校验比特 2 三路数据流分为前后两个部分。系统比特 (校验比特 1 和校验比特 2) 的第一部分对应比特收集的交织矩形前 N_c 列的系统比特 (校验比特 1 和校验比特 2)，系统比特 (校验比特 1 和校验比特 2) 的第二部分对应比特收集的交织矩形 $N_{col}-N_c$ 列的系统比特 (校验比特 1 和校验比特 2)。

[0076] 4、对于三路数据流的第一部分，根据比特收集参数 (N_{col}，N_c 和 N_r) 计算出比特收集交织矩形第一部分数据块中的元素 $m_{0,y}$，$m_{1,y}$，$m_{2,y}$，$m_{3,y}$，$m_{4,y}$，$m_{5,y}$，$m_{6,y}$，$m_{7,y}$ 分别对应的数据源 (系统比特，校验比特 1 或校验比特 2)，因为同一个数据序列来自同一个数据源，确定了八

个数据序列分别对应的数据源就可以确定每个数据序列中连续的比特对应的源头数据的序号。因此可以方便地计算出三路速率匹配输出的数据流的每一个比特对应的数据序列，即可以确定每个比特对应的存放比特收集结果的存储单元。

[0077] 5、因为 y 值的大小次序与速率匹配输出先后次序相同，并且同一序列的数据来自同一个源头，因此将输出到同一序列的比特数据进行拼接存储，可以避免了使用位寻址存储单元。本示例采用 8 位宽的存储单元，当速率匹配输出到同一序列的数据达到 8 个比特后一并写入存储单元保存。速率匹配后的数据源的第一部分数据全部输出完成，将每个序列的最后一笔数据（可能不足 8 个比特，比特个数标记为 n_x，n_x 大于 0，且小于等于 8，x 表示数据的序列号，x 取值范围从 0 到 7）写入存储单元的同时也暂存到一组寄存器。

[0078] 6、对于第二部分数据，同样根据比特收集参数（N_{col}，N_c 和 N_r）计算出比特收集交织矩形第二部分数据块中的元素 $m_{0,y}$，$m_{1,y}$，$m_{2,y}$，$m_{3,y}$，$m_{4,y}$，$m_{5,y}$，$m_{6,y}$，$m_{7,y}$ 分别对应的数据源（系统比特，校验比特 1 或校验比特 2）。x 序列的第二部分数据前 $8-n_x$ 个比特保存到寄存器，从 $9-n_x$ 比特开始按照步骤 5 中类似的方法满 8 个比特后写入步骤 4 中计算出的对应的存储单元，直到速率匹配后的 3 个数据流的第二部分的数据全部输出完成。

[0079] 7、最后根据比特收集参数（N_{col}，N_c 和 N_r）计算出每个序列在 N_c 左右边界既包含第一部分数据又包含第二部分数据的存储单元的地址。将第一部分数据暂存到寄存器的 n_x 个比特和第二部分数据暂存到寄存器的 $8-n_x$ 个比特拼接成一个 8 比特数据更新到该地址。

[0080] 8、做比特加扰和交织运算处理时，将八个存储单元当作一块 64 位存储器。E-DCH 16QAM 调制方式下的交织处理采用如图 10 所示的两个相同结构的 R2×30 交织器：数据流 $s_{(4*i+1)}$，$s_{(4*i+2)}$ 送入交织器一，数据流 $s_{(4*i+3)}$，$s_{(4*i+4)}$ 送入交织器二，输出结果分别命名为 $v_{(4*i+1)}$，$v_{(4*i+2)}$ 和 $v_{(4*i+3)}$，$v_{(4*i+4)}$（其中，i 的取值范围从 0 到 (R/4-1)），按 v 的序号从小到大排列即得到交织后的数据流．根据本装置的数据存储方式，两个交织器所需的数据存储在同一个地址，因此两个交织器采用并行处理，从当前交织运算所需数据的地址读出数据，共获得 64 比特数据，按照 x 序号的次序从 0 到 7 排列，y 值从 y 到 y+7 排列，即还原了比特收集后的数据流：$m_{0,y}$，$m_{1,y}$，$m_{2,y}$，$m_{3,y}$，$m_{4,y}$，$m_{5,y}$，$m_{6,y}$，$m_{7,y}$。。。。 $m_{0,y+7}$，$m_{1,y+7}$，$m_{2,y+7}$，$m_{3,y+7}$，$m_{4,y+7}$，$m_{5,y+7}$，$m_{6,y+7}$，$m_{7,y+7}$。

[0081] 10、将通过交织地址获得的 64 位数据并经过还原的数据流并行做比特加扰处理和物理信道映射等后级处理。

[0082] 所述方法中步骤 4、5 和 6，其第一部分数据流和第二部分数据流只在同一数据源有先后次序，对于不同数据源先后次序不能确定。同一个序列（即同一个存储单元）中的第一部分数据流和第二部分数据流可能来自不同数据源，因此先后次序也不能确定。因而在存储数据时可能会遇到第一部分数据和第二部分数据同时写入相同存储单元的情况。本发明对此类操作做了保护，通过分别控制两部分数据存储，并在上层增加仲裁逻辑确保两部分数据都可以保存到存储单元。

[0083] QPSK 调制方式下的比特收集与上述 16QAM 调制方式下类似，但较 16QAM 简单，只需要将数据分为四个序列，奇数列的第一行，奇数列第二行，偶数列的第一行和偶数列的第二行，使用四个存储单元。同时 QPSK 调制方式下交织也只需要用到一个 R2×30 交织器，整个运算处理是 16QAM 调制方式下的一个子集。

[0084] 通过上述的文字表述并结合附图可以看出，采用本发明后，不需要增加速率匹配

模块与比特收集模块之间的缓存就能够实现混合自动重传请求比特收集和交织，通过改变现有技术的交织所用的存储器的结构，在比特收集结果存储和后级交织输入适当增加控制逻辑，完全避免了速率匹配到比特收集间缓存的使用，减小了电路面积，提高了硬件的工作效率。同时新的比特收集结果的存储结构也方便了后级两路交织处理的并行实现，进一步减小了整个物理过程的延迟时间，提高了 TD-HSUPA 系统的处理能力。

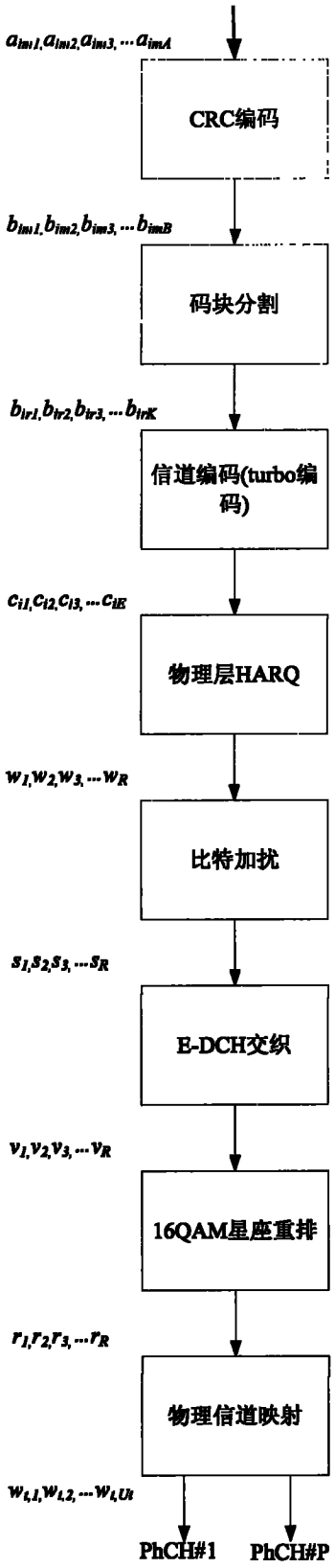

$a_{im1}, a_{im2}, a_{im3}, \cdots a_{imA}$
CRC编码
$b_{im1}, b_{im2}, b_{im3}, \cdots b_{imB}$
码块分割
$b_{ir1}, b_{ir2}, b_{ir3}, \cdots b_{irK}$
信道编码(turbo编码)
$c_{i1}, c_{i2}, c_{i3}, \cdots c_{iE}$
物理层HARQ
$w_1, w_2, w_3, \cdots w_R$
比特加扰
$s_1, s_2, s_3, \cdots s_R$
E-DCH交织
$v_1, v_2, v_3, \cdots v_R$
16QAM星座重排
$r_1, r_2, r_3, \cdots r_R$
物理信道映射
$w_{t,1}, w_{t,2}, \cdots w_{t,Ut}$
PhCH#1
PhCH#P

图 1

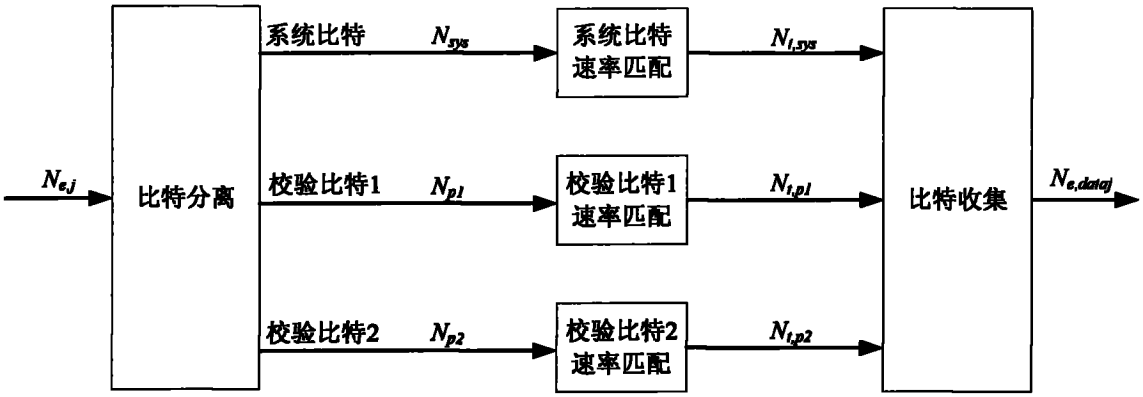


图 2

S	S	…	S	S	S	S
S	S	…	S	S	S	S
S	S	…	S	S	S	S
S	S	…	S	S	S	S

图 3(a)

S	S	…	S	S	S	…
S	S	…	S	S	S	…
S	S	…	S	S	S	…
S	S	…	S	P2	P1	…

图 3(b)

S	S	…	S	S	S	…
S	S	…	S	S	S	…
S	S	…	S	S	S	…
P2	P1	…	P1	P2	P1	…

图 3(c)

S	S	…	S	S	S	…
S	S	…	S	S	S	…
S	S	…	S	P1	P1	…
P2	P1	…	P2	P2	P2	…

图 3(d)

S	S	…	S	S	S	…
S	S	…	S	S	S	…
S	S	…	S	P2	P2	…
P2	P1	…	P1	P1	P1	…

图 3(e)

S	S	…	S	S	S	…
S	S	…	S	S	S	…
P2	P2	…	P2	P2	P2	…
P1	P1	…	P1	P1	P1	…

图 3(f)

S	S	…	S	S	S	…
S	S	…	S	P2	P1	…
P2	P2	…	P2	P1	P2	…
P1	P1	…	P1	P2	P1	…

图 3(g)

S	S	…	S	S	S	…
P2	P1	…	P2	P1	P2	…
P1	P2	…	P1	P2	P1	…
P2	P1	…	P2	P1	P2	…

图 3(h)

S	S	…	S	P1	P1	…
P2	P1	…	P2	P2	P2	…
P1	P2	…	P1	P1	P1	…
P2	P1	…	P2	P2	P2	…

图 3(i)

S	S	…	S	P2	P2	…
P2	P1	…	P1	P1	P1	…
P1	P2	…	P2	P2	P2	…
P2	P1	…	P1	P1	P1	…

图 3(j)

P2	P2	…	P2	P2	P2	…
P1	P1	…	P1	P1	P1	…
P2	P2	…	P2	P2	P2	…
P1	P1	…	P1	P1	P1	…

图 3(k)

S	S	…	S	S	S	…
S	S	…	S	S	S	…

图 3(l)

S	S	…	S	S	S	…
S	S	…	S	P2	P1	…

图 3(m)

S	S	…	S	S	S	…
P2	P1	…	P2	P1	P2	…

图 3(n)

S	S	…	S	P1	P1	…
P2	P1	…	P2	P2	P2	…

图 3(o)

S	S	…	S	P2	P2	…
P2	P1	…	P1	P1	P1	…

图 3(p)

P2	**P2**	…	**P2**	**P2**	**P2**	…
P1	**P1**	…	**P1**	**P1**	**P1**	…

图 3(q)

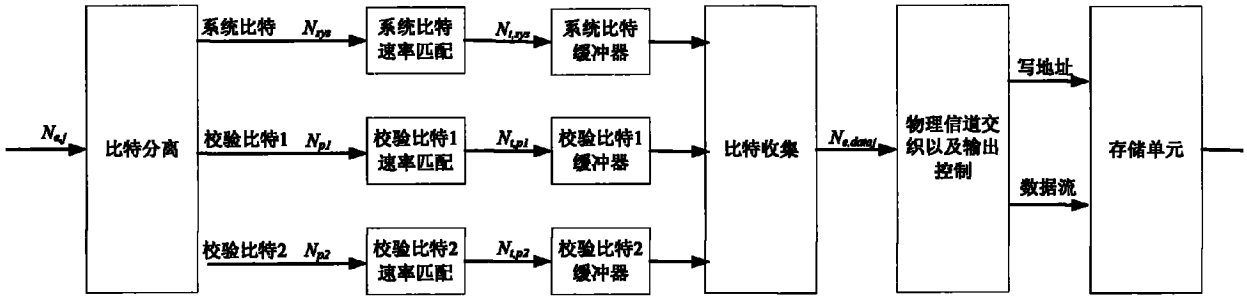


图 4(a)

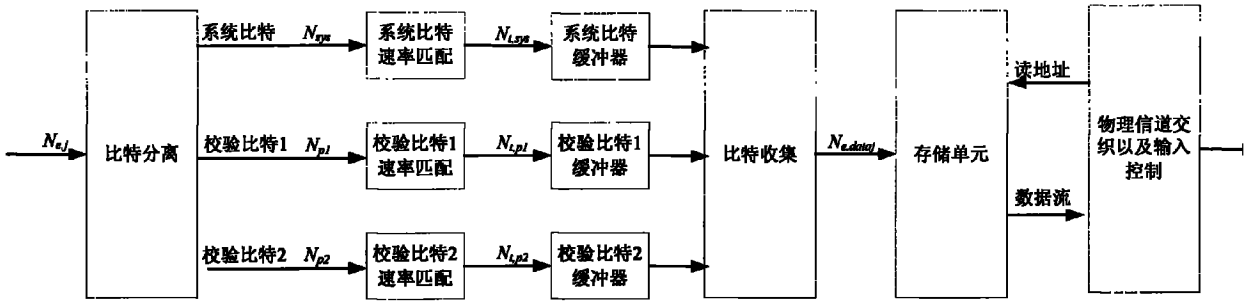


图 4(b)

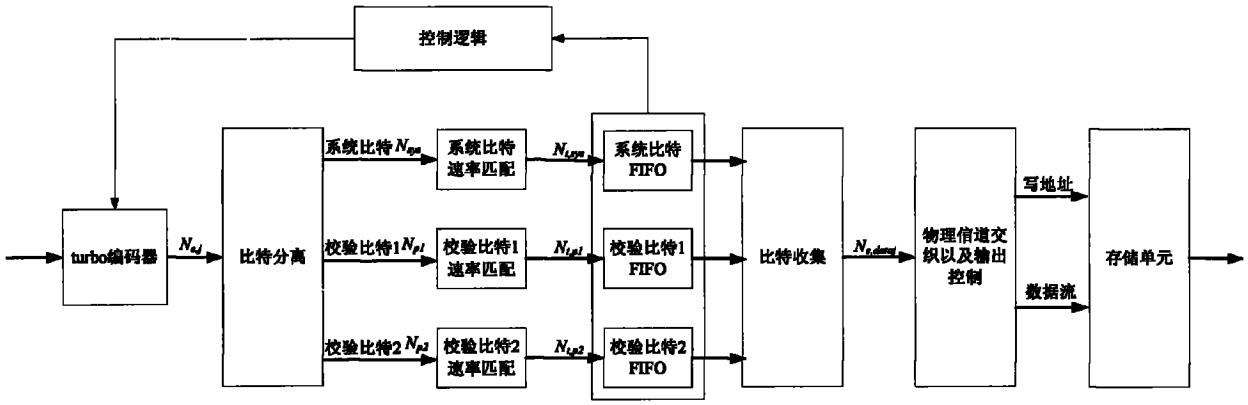


图 5

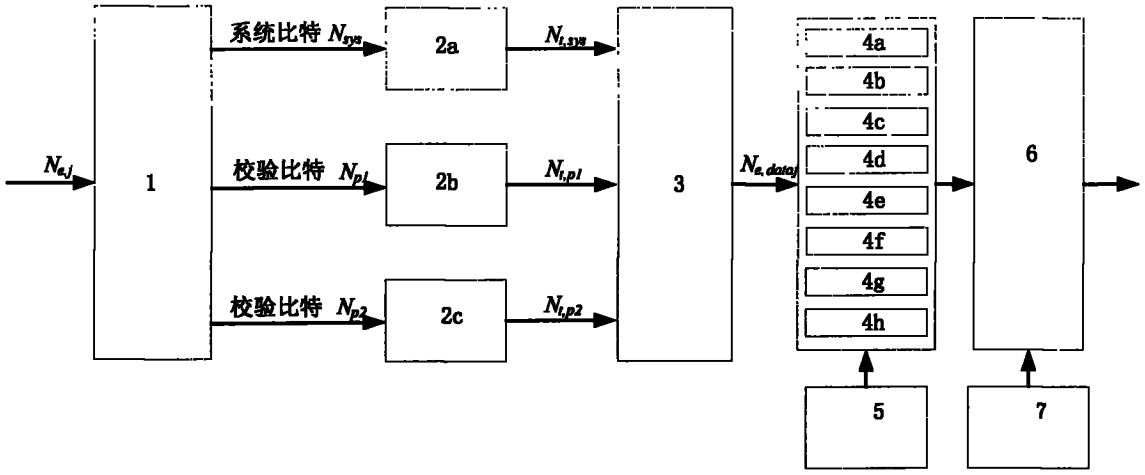


图 6

m00	m40	m01	m41	…	m06	m46	m07	m47	m08	m48	…	…	…	…	…
m10	m50	m11	m51	…	m16	m56	m17	m57	m18	m58	…	…	…		…
m20	m60	m21	m61	…	m26	m66	m27	m67	m28	m68	…	…	…		…
m30	m70	m31	m71	…	m36	m76	m37	m77	m38	m78	…	…	…	…	…

图 7

m07 ……m01,m00
m015 ……m09,m08
……

图 8(a)

m17 ……m11,m10
m115 ……m19,m18
……

图 8(b)

m27 ……m21,m20
m215 ……m29,m28
……

图 8(c)

m37 ……m31,m30
m315 ……m39,m38
……

图 8(d)

m47 ……m41,m40
m415 ……m49,m48
……

图 8(e)

m57 ……m51,m50
m515 ……m59,m58
……

图 8(f)

m67 ……m61,m60
m615 ……m69,m68
……

图 8(g)

m77 ……m71,m70
m715 ……m79,m78
……

图 8(h)

N_c

第一部分数据							第二部分数据				
S	S	S	S	…	S	S	P2	P2	P2	P2	…
P2	P1	P2	P1	…	P2	P1	P1	P1	P1	P1	…
P1	P2	P1	P2	…	P1	P2	P2	P2	P2	P2	…
P2	P1	P2	P1	…	P2	P1	P1	P1	P1	P1	…

图 9(a)

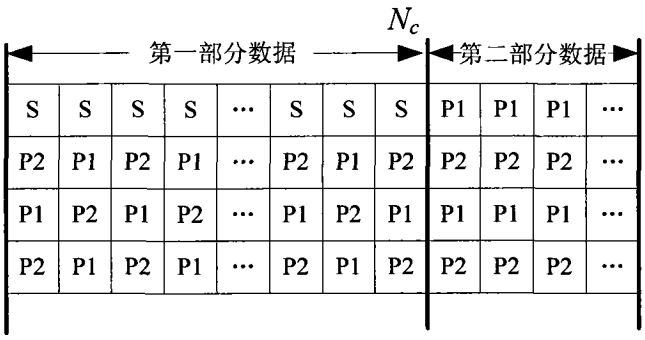

N_c
第一部分数据
第二部分数据
S S S S … S S S P1 P1 P1 …
P2 P1 P2 P1 … P2 P1 P2 P2 P2 P2 …
P1 P2 P1 P2 … P1 P2 P1 P1 P1 P1 …
P2 P1 P2 P1 … P2 P1 P2 P2 P2 P2 …

图 9(b)

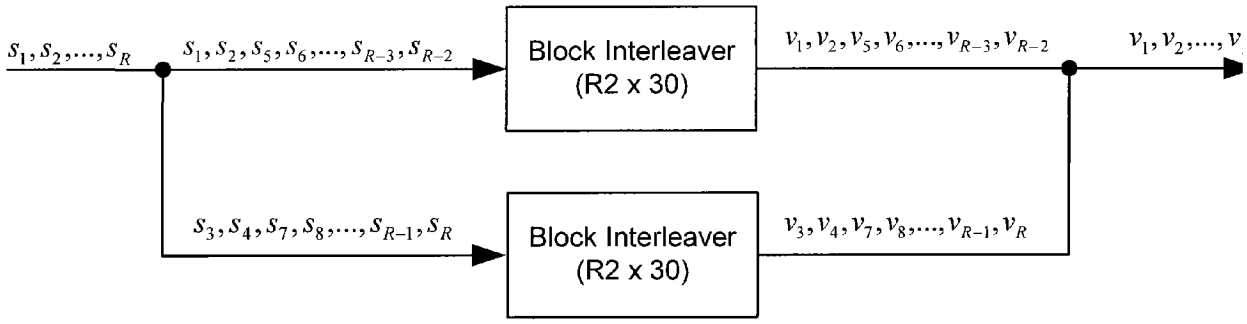

$s_1, s_2, \ldots, s_R$
$s_1, s_2, s_5, s_6, \ldots, s_{R-3}, s_{R-2}$
Block Interleaver (R2 x 30)
$v_1, v_2, v_5, v_6, \ldots, v_{R-3}, v_{R-2}$
$v_1, v_2, \ldots, v$
$s_3, s_4, s_7, s_8, \ldots, s_{R-1}, s_R$
Block Interleaver (R2 x 30)
$v_3, v_4, v_7, v_8, \ldots, v_{R-1}, v_R$

图 10