(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103812608 B
(45)授权公告日 2017.04.26

(21)申请号 201310736738.X

(22)申请日 2013.12.26

(65)同一申请的已公布的文献号
申请公布号 CN 103812608 A

(43)申请公布日 2014.05.21

(73)专利权人 西安交通大学
地址 710049 陕西省西安市碑林区咸宁西路28号

(72)发明人 张超 陈亚伟

(74)专利代理机构 西安通大专利代理有限责任公司 61200
代理人 陆万寿

(51)Int.Cl.
H04L 1/00(2006.01)
H04L 29/06(2006.01)

(56)对比文件
CN 102075467 A,2011.05.25,
US 2011/0145549 A1,2011.06.16,
CN 102215199 A,2011.10.12,
CN 102790656 A,2012.11.21,
CN 102790999 A,2012.11.21,

审查员 张翔

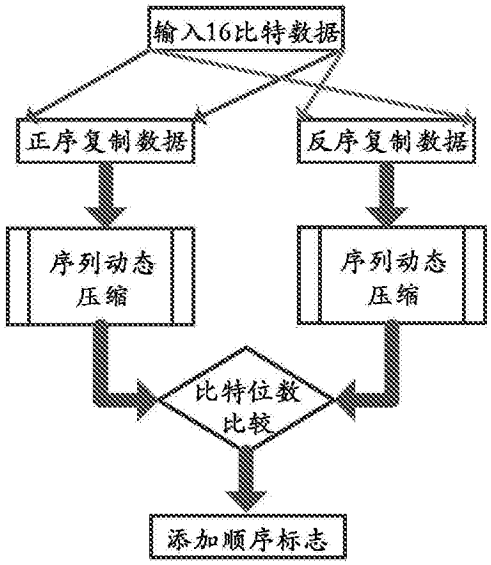
权利要求书1页 说明书6页 附图1页

(54)发明名称

一种IQ数据压缩方法和系统

(57)摘要

本发明公开了一种IQ数据压缩方法和系统，该方法包括：在发送端，对于每组数据先将其转换为原码形式的Lbit的二进制数，保留其符号位(即第一位)，同时在首位处添加一位作为正反序列标志位，对余下的(L-1)bit进行处理，根据自身特点将此数据分为若干组m，实行合理舍弃，得到压缩数据，该方法以并行方式，正反序列同时执行该算法，以使其到达EVM及压缩比的最优；在接收端，将接收到的压缩数据根据各组的特点，进行补零，完成相应的数据恢复。本发明一种IQ数据压缩方法和系统，其可以实现多载波IQ数据压缩同步进行，根据正反序列的压缩效果传输压缩较明显的一路数据，不仅提高了压缩效率，而且易于硬件实现。



CN 103812608 B

1.一种IQ数据压缩方法，其特征在于，包括以下步骤：

1)在发送端，先将IQ数据转换为原码形式的Lbit二进制数，保留该Lbit二进制数符号位，在正序列Lbit二进制数的第一位添加标示符0作为标志位，在反序列Lbit二进制数的第一位添加标示符1作为标志位，得到两组并行的带有标志位的正序列Lbit二进制数和反序列Lbit二进制数；

2)根据IQ数据的特点，除符号位和标志位外，将增加标志位后的正序列余下的(L-1)bit二进制数分为m个小组，将增加标志位后的反序列余下的(L-1)bit二进制数分为n个小组，根据每个小组的特征，分别对分组后的正序列Lbit二进制数和分组后的反序列Lbit二进制数舍弃其若干个尾部比特，其中，m、n均为正整数，且有1＜m＜L-1，1＜n＜L-1；

3)发送端比较舍弃尾部比特后的正序列Lbit二进制数和舍弃尾部比特后的反序列Lbit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列Lbit二进制数或者舍弃尾部比特后的反序列Lbit二进制数作为传输数据；

4)接收端根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列Lbit二进制数还是反序列Lbit二进制数，实现压缩数据的解压。

2.根据权利要求1所述的一种IQ数据压缩方法，其特征在于，步骤3)中，根据舍弃尾部比特后的正序列Lbit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到正序列Lbit二进制数的压缩数据；根据舍弃尾部比特后的反序列Lbit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到反序列Lbit二进制数的压缩数据。

3.一种IQ数据压缩系统，其特征在于，包括发送端和接收端；其中，

发送端，用于将IQ数据转换为原码形式的Lbit二进制数，保留该Lbit二进制数符号位，在正序列Lbit二进制数的第一位添加标示符0作为标志位，在反序列Lbit二进制数的第一位添加标示符1作为标志位，得到两组并行的带有标志位的正序列Lbit二进制数和反序列Lbit二进制数；除符号位和标志位外，用于将增加标志位后的正序列(L-1)bit二进制数分为m个小组，将增加标志位后的反序列(L-1)bit二进制数分为n个小组，能够对分组后的正序列Lbit二进制数和分组后的反序列Lbit二进制数舍弃其若干个尾部比特；用于比较舍弃尾部比特后的正序列Lbit二进制数和舍弃尾部比特后的反序列Lbit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列Lbit二进制数或者舍弃尾部比特后的反序列Lbit二进制数作为传输数据；

接收端，用于根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列Lbit二进制数还是反序列Lbit二进制数，实现压缩数据的解压。

4.根据权利要求3所述的一种IQ数据压缩系统，其特征在于，所述发送端，用于动态调整正反序列的分组数。

一种IQ数据压缩方法和系统

【技术领域】

[0001] 本发明涉及移动通信技术领域，特别涉及一种IQ数据压缩方法和系统。

【背景技术】

[0002] 在移动通信系统中，基站包括基带处理单元(BBU)与射频拉远单元(RRU)，BBU和RRU之间采用光纤连接，双向传输IQ数据。IQ数据即为基带数字信号。基带数字信号包括I路信号和Q路信号。

[0003] 在TDD-LTE系统的基站中，BBU与RRU之间采用IR(Interface between the BBU and the RRU)协议来进行IQ数据的传输，随着空口的传输数据不断提高，造成IR传输的压力和成本不断上升，业界有多个厂家提出了多种不同的方法，来压缩IQ数据传输的位宽。

[0004] 目前已知的IQ数据压缩方案有线性压缩和非线性压缩。非线性压缩，如A律压缩方案等相对线性压缩来说实现比较复杂。目前的一种线性压缩方案如下：

[0005] 对下行IQ数据进行分组，将连续m个IQ数据划分为一组，并分别获取各组数据中I路信号和Q路信号的数值最大数据；截取本组数据中所述数值最大数据的从含1的比特位开始的连续比特高位有效数据以及符号位，并删除所述数值最大数据中剩余的低比特位数据；截取本组数据中其他m-1个数据的所述连续n比特高位有效数据以及符号位，并删除所述其他m-1个数据中所述剩余的低比特位数据；根据删除的低比特位数据的位数确定本组的压缩因子，并发送所述压缩因子和压缩后的IQ数据。在解压缩端根据压缩因子对数据进行还原。

[0006] 这种线性压缩方案依据每组数据中I路信号和Q路信号的模的最大值，根据其高位零的个数判断移位因子，需要对I路信号和Q路信号分开计算压缩，并且对于包含负数的小信号压缩损失明显。

【发明内容】

[0007] 本发明的目的在于克服上述现有技术中的缺陷，提供了一种IQ数据压缩方法和系统，该方法可以实现多载波IQ数据压缩同步进行，根据正反序列的压缩效果传输压缩较明显的一路数据，不仅提高了压缩效率，而且易于硬件实现。

[0008] 为达到上述目的，本发明的技术方案是这样实现的：

[0009] 一种IQ数据压缩方法，包括以下步骤：

[0010] 1)在发送端，先将IQ数据转换为原码形式的Lbit二进制数，保留该Lbit二进制数符号位，在正序列Lbit二进制数的第一位添加标示符0作为标志位，在反序列Lbit二进制数的第一位添加标示符1作为标志位，得到两组并行的带有标志位的正序列Lbit二进制数和反序列Lbit二进制数；

[0011] 2)根据IQ数据的特点，除符号位和标志位外，将增加标志位后的正序列余下的(L-1)bit二进制数分为m个小组，将增加标志位后的反序列余下的(L-1)bit二进制数分为n个小组，根据每个小组的特征，分别对分组后的正序列Lbit二进制数和分组后的反序列Lbit

二进制数舍弃其若干个尾部比特，其中，m、n均为正整数，且有1＜m＜L-1，1＜n＜L-1；

[0012] 3)发送端比较舍弃尾部比特后的正序列Lbit二进制数和舍弃尾部比特后的反序列Lbit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列Lbit二进制数或者舍弃尾部比特后的反序列Lbit二进制数作为传输数据；

[0013] 4)接收端根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列Lbit二进制数还是反序列Lbit二进制数，实现压缩数据的解压。

[0014] 本发明进一步改进在于，步骤3)中，根据舍弃尾部比特后的正序列Lbit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到正序列Lbit二进制数的压缩数据；根据舍弃尾部比特后的反序列Lbit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到反序列Lbit二进制数的压缩数据。

[0015] 一种IQ数据压缩系统，包括发送端和接收端；其中，

[0016] 发送端，用于将IQ数据转换为原码形式的Lbit二进制数，保留该Lbit二进制数符号位，在正序列Lbit二进制数的第一位添加标示符0作为标志位，在反序列Lbit二进制数的第一位添加标示符1作为标志位，得到两组并行的带有标志位的正序列Lbit二进制数和反序列Lbit二进制数；除符号位和标志位外，用于将增加标志位后的正序列(L-1)bit二进制数分为m个小组，将增加标志位后的反序列(L-1)bit二进制数分为n个小组，能够对分组后的正序列Lbit二进制数和分组后的反序列Lbit二进制数舍弃其若干个尾部比特；用于比较舍弃尾部比特后的正序列Lbit二进制数和舍弃尾部比特后的反序列Lbit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列Lbit二进制数或者舍弃尾部比特后的反序列Lbit二进制数作为传输数据；

[0017] 接收端，用于根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列Lbit二进制数还是反序列Lbit二进制数，实现压缩数据的解压。

[0018] 本发明进一步改进在于，所述发送端，用于动态调整正反序列的分组数。

[0019] 相对于现有技术，本发明具有如下技术效果：

[0020] 本发明一种IQ数据压缩方法和系统，该方法采用的是正反序列并行压缩，也即从两个不同的方向上同时进行压缩，最终比较二者之间的压缩效果，取压缩效果明显的那组压缩数据，本发明权衡了低幅度值的样点和高幅度值的样点的特点，使二者都可以达到一个较高的压缩效果，且保证了解压后的数据误差较低，实现了高压缩比和低误码率的双重优势。

【附图说明】

[0021] 图1为本发明IQ数据压缩方法和系统的流程图。

【具体实施方式】

[0022] 下面结合附图和具体实施例对本发明作进一步说明。

[0023] 参见图1所示，以每组数据长度为16bit为例，本发明IQ数据压缩方法，包括以下步骤：

[0024] 1)在发送端，先将IQ数据转换为原码形式的16bit二进制数，保留该16bit二进制数符号位，在正序列16bit二进制数的第一位添加标示符0作为标志位，在反序列16bit二进

制数的第一位添加标示符1作为标志位，得到两组并行的正序列16bit二进制数和反序列16bit二进制数；

[0025] 2)根据IQ数据的特点，除符号位和标志位外，将增加标志位后的正序列16bit二进制数分为m个小组，将增加标志位后的反序列16bit二进制数分为n个小组，根据每个小组的特征，分别对分组后的正序列16bit二进制数和分组后的反序列16bit二进制数舍弃其若干个尾部比特，其中，m、n均为正整数，且有1＜m＜15，1＜n＜15；

[0026] 3)发送端比较舍弃尾部比特后的正序列16bit二进制数和舍弃尾部比特后的反序列16bit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列16bit二进制数或者舍弃尾部比特后的反序列16bit二进制数作为传输数据；

[0027] 4)接收端根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列16bit二进制数还是反序列16bit二进制数，实现压缩数据的解压。

[0028] 其中，上述步骤3)中，根据舍弃尾部比特后的正序列16bit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到正序列16bit二进制数的压缩数据；根据舍弃尾部比特后的反序列16bit二进制数的分组情况，对其每个IQ数据的I数据和Q数据进行截位，得到反序列16bit二进制数的压缩数据。

[0029] 一种IQ数据压缩系统，包括发送端和接收端；其中，

[0030] 发送端，用于将IQ数据转换为原码形式的Lbit二进制数，保留该Lbit二进制数符号位，在正序列Lbit二进制数的第一位添加标示符0作为标志位，在反序列Lbit二进制数的第一位添加标示符1作为标志位，得到两组并行的正序列Lbit二进制数和反序列Lbit二进制数；除符号位和标志位外，用于将增加标志位后的正序列Lbit二进制数分为m个小组，将增加标志位后的反序列L-1bit二进制数分为n个小组，能够对分组后的正序列Lbit二进制数和分组后的反序列Lbit二进制数舍弃其若干个尾部比特；用于比较舍弃尾部比特后的正序列Lbit二进制数和舍弃尾部比特后的反序列Lbit二进制数压缩后的比特个数，取比特个数少的舍弃尾部比特后的正序列Lbit二进制数或者舍弃尾部比特后的反序列Lbit二进制数作为传输数据；另外，发送端算法根据 $EVM=\sqrt{\frac{\sum_{n=1}^{L}|Z[n]-I[n]|^2}{\sum_{n=1}^{L}|I[n]|^2}}\times 100\%$ 计算公式，当EVM小于1％时，选择最大的压缩比；此外需要说明的是，实现设计的压缩比和误差向量幅度的折中。

[0031] 接收端，用于根据接收的压缩数据，补充对其舍弃尾部比特相应长度的零比特，判断其是正序列Lbit二进制数还是反序列Lbit二进制数，发送端可以实现压缩数据的解压。

[0032] 上述压缩算法的基本原理为：

[0033] 对于每组数据先将其转换为原码形式的16bit的二进制数，保留其符号位(即第一位)，对余下的15bit进行处理，将其分为若干小组(如N组)，每组所对应的bit分别为M_1，M_2，…M_N，根据每组的具体情况，舍弃末尾的不同比特数，

[0034] $$\sum_{i=1}^{N}\mathrm{size}(M_i)=15$$

[0035] 其中size()表示计算每个小组的比特个数(如下图)。然后根据分组的数值大小对原始数据进行不同比特位数的舍弃,从而实现不同幅度信号的动态压缩。

[0036]

符号 1bit	第一组 M_1 比特	第二组 M_2 比特	第三组 M_3 比特	.....	第 N 组 M_N 比特

[0037] 压缩后数据格式为

[0038]

顺序标示符1bit	符号位1bit	压缩后数据比特

[0039] 请注意,顺序标示符为1时表示此压缩结果为反序数据压缩所得,当标示为0时表示正序数据压缩所得。此标志位由本算法特别定义。

[0040] 压缩算法具体步骤:

[0041] 设某一子载波第i个I(或Q)路的16比特原始数据是x_i=x_i[1:16]。(Mi)10表示Mi分组对应的10进制数。n_1,n_2,n_3.....n_N代表不同分组需要舍弃的比特个数,Num_i表示所第i个数据压缩后剩比特数,y_i表示压缩后的数据。为了解压时一一映射,我们要求 $0 \le n_i \le 15-\sum_{k=1}^{i} M_k$, $1 \le i \le N-1$ 。

[0042] 序列动态压缩子程序

[0043] 第i个16比特数据的动态压缩过程为:

[0044] if $(M_1)_{10}>0$

[0045] Num_i=16-n_1;

[0046] y_i=x_i[1:Num_i];

[0047] 原数据压缩为

[0048]

符号位	x_i[2:Num_i]

[0049] elseif $(M_2)_{10}>0$

[0050] Num_i=16-n_2

[0051] y_i=x_i[1:Num_i];

[0052] 原数据压缩为

[0053]

符号位	x_i[2:Num_i]

[0054] elseif $(M_2)_{10}>0$

[0055] Num_i=16-n_3;

[0056] y_i=x_i([1:Num_i]);

[0057]

符号位	x_i[2:Num_i]

[0058] else if $(M_{N-1})_{10}>0$

[0059] Num_i=16-n_{N-1};

[0060] y_i=x_i([1:Num_i]);

[0061] 原数据压缩为

[0062]

符号位	x_i[2:Num_i]

[0063] end

[0064] 注意：判断$(M_i)_{10}$大于零可以直接通过M_i分组进行比特或运算，例如，分组为1101，则判断过程为1或1或0或1＝1，便于硬件实现。对于低幅度样点，该子程序压缩效果较差，例如$(0001)_{16}$。

[0065] 正反序列并行动态压缩算法

[0066] 根据第i个16比特数据x_i[1:16]，产生正序序列s_i[1:16]＝x_i[1:16]，同时产生反序序列c_i，其中符号位不变c_i[1]＝x_i[1]，数据位反序即c_i[2:16]＝x_i[16:2]。例如，1110 0100 1010 0000的反序序列为1000001010010011.然后将s_i和c_i分别输入序列动态压缩子程序，并行地进行数据压缩。之后对两个压缩后的数据比特数进行比较，保留比特数较少的结果。最后在保留的结果前添加顺序标示符。算法流程图如图1所示：

[0067] 上述解压算法具体步骤为：

[0068] 首先提取顺序标示位，然后将符号位以及压缩结果输入解压缩子程序。解压缩子程序具体如下：

[0069] if $(M_1)_{10}>0$

[0070] 将该数据后面自动补充n_1个0

[0071] else if $(M_2)_{10}>0$

[0072] 将该数据后面自动补充n_2个0

[0073] else if $(M_2)_{10}>0$

[0074] 将该数据后面自动补充n_3个0

[0075] else将该数据后面自动补充n_N个0；

[0076] end

[0077] 解压缩后，如果顺序标示位为1，则将解压缩结果的数据比特反序。

[0078] 注意：

[0079] 这样对于不同幅度的样点进行不同比特位的舍去，也就是进行不同的误差控制和压缩比控制，可以灵活的实现需要的EVM和压缩率。尤其是当幅度值较小的样点，完全可以舍去较多位数比特，例如n_{N-1}＝8.

[0080] 另外，全部算法模块采用二进制计算，便于在硬件实现。

[0081] 数据源文件格式说明：

[0082] 为了方便对算法进行评估，我们在这里提供了两种测试场景，每个场景的数据存放于一个文件中：

[0083] 1.低吞吐率场景

[0084] a.数据包含两个载波

[0085] b.采样率30.72MSPS

[0086] c.数据长度10ms，即对于一个载波有30.72M*10m＝307,200个采样点(文件总采样点为307,200*2*2＝1,228,800个)

[0087] d.调制方式QPSK

[0088] e.网络平均负载50％

[0089] 2.高吞吐率场景

[0090] a.数据包含八个载波

[0091] b.采样率19.2MSPS

[0092] c.数据长度10ms，即对于一个载波有19.2M*10m＝192,000个采样点(文件总采样点为192,000*2*8＝3,072,000个)

[0093] d.调制方式64QAM

[0094] e.网络平均负载100％

[0095] 具体来说，文件中每行表示一个载波(复数)的I(实部)或Q(虚部)数据，I/Q数据交替排列，所有载波依次循环出现。

[0096] 下面以高吞吐率场景为例说明数据源文件中I、Q数据的排列格式(低吞吐率场景只包含2载波数据)

[0097]

行号	文件内容	描述
1	F9B6	载波1的I数据(第1个采样时间)
2	05DA	载波1的Q数据(第1个采样时间)
3	F9B6	载波2的I数据(第1个采样时间)
4	05DA	载波2的Q数据(第1个采样时间)
5	0305	载波3的I数据(第1个采样时间)
6	FD31	载波3的Q数据(第1个采样时间)
…	…	…
15	08D3	载波8的I数据(第1个采样时间)
16	072A	载波8的Q数据(第1个采样时间)
17	FC99	载波1的I数据(第2个采样时间)
18	FED6	载波1的Q数据(第2个采样时间)
19	FC99	载波2的I数据(第2个采样时间)
20	FED6	载波2的Q数据(第2个采样时间)
…	…	…

[0098] 每行中的数据为16比特有符号整数，其中最高位为符号位，补码表示，采用十六进制表示。

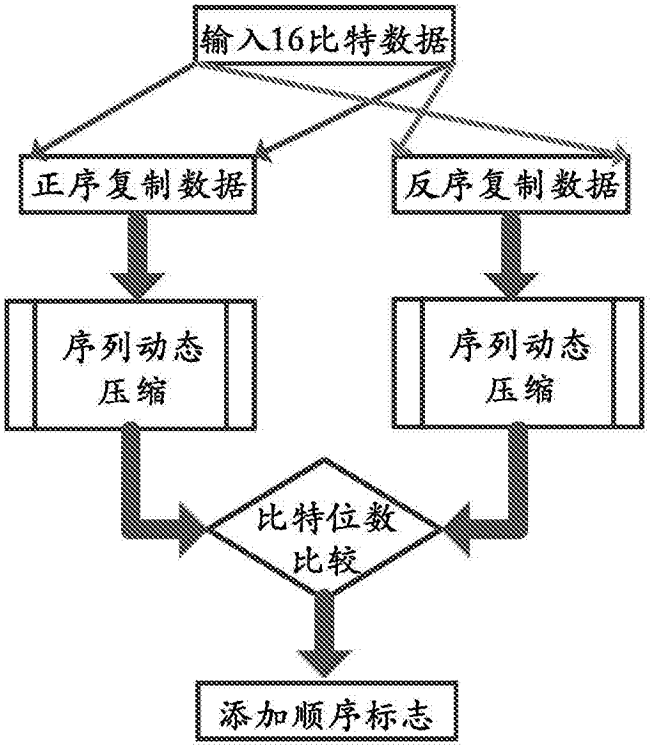


图1