



(12) 发明专利申请

(10) 申请公布号 CN 101969311 A

(43) 申请公布日 2011. 02. 09

(21) 申请号 201010297874. X

(22) 申请日 2010. 09. 29

(71) 申请人 航天恒星科技有限公司
地址 100086 北京市海淀区知春路 82 号院

(72) 发明人 张拯宁 战勇杰

(74) 专利代理机构 中国航天科技专利中心
11009

代理人 安丽

(51) Int. Cl.

H03M 13/41 (2006. 01)

H03M 13/27 (2006. 01)

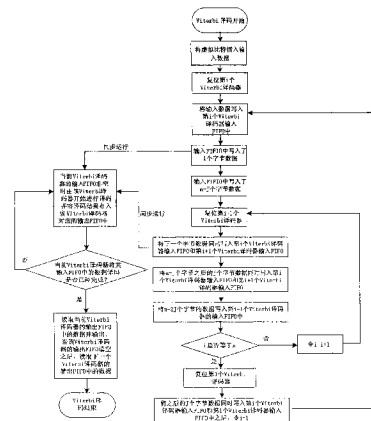
权利要求书 1 页 说明书 5 页 附图 2 页

(54) 发明名称

一种高速并行分段交错维特比译码方法

(57) 摘要

一种高速并行分段交错维特比 (Viterbi) 译码方法, 通过采用多级并行分段交错 Viterbi 译码方法, 使译码速率大大提高, 通过在虚拟比特插入时同步生成虚拟比特指示信号, 不仅能够适应 CCSDS 标准规定的各种卷积编码打孔模式, 而且可以方便的扩展为其它打孔模式, 通过采用模块化设计, 在需要时可以进一步通过增加并行度提高性能, 本发明通过大量软件仿真实验和实际测试结果表明, 使用本发明高速并行分段交错 Viterbi 译码方法可使 Viterbi 译码速率达到 800Mbit/s, 优于目前文献中发表的同类型译码器的性能指标。



1. 一种高速并行分段交错维特比译码方法,其特征在于步骤如下:

(1) 将虚拟比特插入输入数据中,之后进入步骤(2);所述输入数据为卷积编码器各支路输出交错排列后经1:8串并转换形成的8bit并行数据;

(2) 复位第*i*个Viterbi译码器之后进入步骤(3),所述*i*为大于等于1且小于等于*m*的自然数,*i*初始化为1,*m*为Viterbi译码器的数量,且 $m \geq 2$;

(3) 将步骤(1)中得到的数据写入第*i*个Viterbi译码器的输入FIFO中,当写入1个字节后启动步骤(7),步骤(7)与步骤(3)并行;当共写入了*n*-*J*个字节之后,进入步骤(4);所述*n*为每个Viterbi译码器单次译码处理的数据长度,且 $n > 6k$,*k*为卷积编码器的约束长度;所述*J*为连续两个Viterbi译码器单次译码处理的数据的交错长度;

(4) 复位第*i*+1个Viterbi译码器之后进入步骤(5);

(5) 将步骤(4)中写入*n*-*J*个字节之后的第一个字节写入第*i*个Viterbi译码器的输入FIFO和第*i*+1个Viterbi译码器的输入FIFO中之后启动步骤(7),步骤(7)与步骤(5)并行;当将步骤(4)中写入*n*-*J*个字节之后的共计*J*个字节的数据同时写入第*i*个Viterbi译码器的输入FIFO和第*i*+1个Viterbi译码器的输入FIFO中之后,再将写入*J*个字节之后的*n*-2*J*个字节的数据第*i*+1个Viterbi译码器的输入FIFO中,之后进入步骤(6);

(6) 判断*i*是否等于*m*,若等于,则复位第1个Viterbi译码器并将步骤(5)中写入*n*-2*J*个字节数据之后的*J*个字节的数据同时写入第*i*个Viterbi译码器的输入FIFO和第1个Viterbi译码器的输入FIFO中,之后令*i*=1,返回步骤(3);若不等于,则将*i*+1赋值给*i*,之后返回步骤(4);

(7) 当前Viterbi译码器的输入FIFO非空时由该Viterbi译码器开始进行译码并将译码结果存入该Viterbi译码器对应的输出FIFO中,当该Viterbi译码器将输入FIFO中的数据译码完成之后进入步骤(8);

(8) 读取当前Viterbi译码器的输出FIFO中的数据并输出,当该Viterbi译码器的输出FIFO读空之后,读取下一个Viterbi译码器的输出FIFO中的数据并输出。

2. 根据权利要求1所述一种高速并行分段交错维特比译码方法,其特征在于,所述步骤(1)中插入虚拟比特按照如下步骤进行:

(a) 根据虚拟比特参数判断是否需要插入虚拟比特,若需要插入,则将输入数据按照打孔图样插入虚拟比特并写入数据输出缓存中,同时,将对应打孔图样的虚拟比特指示信号同步写入指示信号缓存中,之后进入步骤(b);若不需要插入,则将输入数据直接输出;所述虚拟比特参数和打孔图样根据所使用的卷积码编码类型确定;

(b) 判断步骤(a)中所述数据输出缓存是否为空,若为空,则返回步骤(a);若不为空,则同时读取数据输出缓存和指示信号缓存中的数据,并输出。

一种高速并行分段交错维特比译码方法

技术领域

[0001] 本发明涉及一种高速并行分段交错维特比 (Viterbi) 译码方法,是针对卷积编码的高速并行 Viterbi 译码方法,主要用于对解调后数据进行高速维特比译码,纠正信道传输过程产生的错误。

背景技术

[0002] 卷积码是一种应用非常广泛的信道编码,主要用于纠正信道传输中产生的随机错误。卷积码主要有三种较好的译码算法:

[0003] (1)1963年由 Massey 提出的门限译码,这是一种利用码代数结构的代数译码方法,类似于分组码中的大数逻辑译码;

[0004] (2)1961年由 Wozencraft 提出,1963年由 Fano 改进的序列译码,这是基于码树图结构上的一种准最佳的概率译码;

[0005] (3)1967年 Viterbi 提出的 Viterbi 算法,这是基于码的网格图 (Trellis) 基础上的一种最大似然算法,是一种最佳概率译码方法。

[0006] 代数译码仅用于简单的卷积码,优点是译码电路简单而且延时小,适用于高速应用;缺点是编码增益一般都不大,而且只能适用于硬判决译码。序列译码和 Viterbi 译码都属于概率译码,由于它们不仅基于码的代数结构,而且利用了信道的统计特性,因而能充分发挥卷积码的特点,使译码错误概率达到很小。Viterbi 译码在码的约束度较小时,比序列译码算法效率更高,速度更快,译码器也更简单。因此 Viterbi 译码得到了广泛应用,特别是在空间通信系统中。CCSDS 建议采用 3 比特量化的 Viterbi 译码。

[0007] Viterbi 译码算法的原理是将接收码和本地生成码比较,基于卷积码的网格图表示法,依据接收到的信息作为先验信息,从所有可能的路径中选择出最可能的序列作为输出。Viterbi 译码器一般由三部分组成:

[0008] (1)BMU(分支度量值单元);

[0009] (2)ACSU(加比选单元);

[0010] (3)SMU(幸存路径存储单元)。

[0011] 如附图 1 所示,由于 Viterbi 译码算法中存在反馈回路,如果只利用现有 Viterbi 译码算法本身的并行结构,Viterbi 译码器的速率将受到限制,目前一般单路 Viterbi 译码器的译码速率最高只能达到 300Mbit/s 左右。要实现更高速率的 Viterbi 译码器,就需要在 Viterbi 译码算法中引入额外的并行结构。但是由于 Viterbi 译码过程是一个连续的译码过程,必须接收连续输入的比特流,在网格图上按照最大似然原理寻找编码序列,这就导致无法直接将输入数据转换为并行数据进行并行译码处理。上述问题造成了在目前技术基础上,Viterbi 译码算法能够实现的最高译码速率有一个上限,这个上限取决于器件水平以及译码算法的优化程度。此时,即使还有足够多的硬件资源可用,但由于无法并行实现,Viterbi 译码器的译码速率也无法再提高。为了解决该难题,通常采取的措施是在发送端采用多个卷积编码器,再结合多进制调制,接收端解调后同样采用多个 Viterbi 译码器进

行译码,这样也相当于进行了并行译码。但是上述措施并没有从根本上解决问题,当单路 Viterbi 译码需要较高的速率时,上述措施将失效,且成本投入变大,浪费了资源。

发明内容

[0012] 本发明的技术解决问题是:克服现有技术的不足,提供了一种高速并行分段交错维特比 (Viterbi) 译码方法,此译码方法提出了单路 Viterbi 译码过程的并行译码方法,大大提高了译码速率。

[0013] 本发明的技术解决方案是:

[0014] 一种高速并行分段交错维特比译码方法,步骤如下:

[0015] (1) 将虚拟比特插入输入数据中,之后进入步骤 (2);所述输入数据为卷积编码器各支路输出交错排列后经 1:8 串并转换形成的 8bit 并行数据;

[0016] (2) 复位第 i 个 Viterbi 译码器之后进入步骤 (3),所述 i 为大于等于 1 且小于等于 m 的自然数, i 初始化为 1, m 为 Viterbi 译码器的数量,且 $m \geq 2$;

[0017] (3) 将步骤 (1) 中得到的数据写入第 i 个 Viterbi 译码器的输入 FIFO 中,当写入 1 个字节后启动步骤 (7),步骤 (7) 与步骤 (3) 并行;当共写入了 $n-J$ 个字节之后,进入步骤 (4);所述 n 为每个 Viterbi 译码器单次译码处理的数据长度,且 $n > 6k$, k 为卷积编码器的约束长度;所述 J 为连续两个 Viterbi 译码器单次译码处理的数据的交错长度;

[0018] (4) 复位第 $i+1$ 个 Viterbi 译码器之后进入步骤 (5);

[0019] (5) 将步骤 (4) 中写入 $n-J$ 个字节之后的第一个字节写入第 i 个 Viterbi 译码器的输入 FIFO 和第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中之后启动步骤 (7),步骤 (7) 与步骤 (5) 并行;当将步骤 (4) 中写入 $n-J$ 个字节之后的共计 J 个字节的数据同时写入第 i 个 Viterbi 译码器的输入 FIFO 和第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中之后,再将写入 J 个字节之后的 $n-2J$ 个字节的数据第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中,之后进入步骤 (6);

[0020] (6) 判断 i 是否等于 m ,若等于,则复位第 1 个 Viterbi 译码器并将步骤 (5) 中写入 $n-2J$ 个字节数据之后的 J 个字节的数据同时写入第 i 个 Viterbi 译码器的输入 FIFO 和第 1 个 Viterbi 译码器的输入 FIFO 中,之后令 $i = 1$,返回步骤 (3);若不等于,则将 $i+1$ 赋值给 i ,之后返回步骤 (4);

[0021] (7) 当前 Viterbi 译码器的输入 FIFO 非空时由该 Viterbi 译码器开始进行译码并将译码结果存入该 Viterbi 译码器对应的输出 FIFO 中,当该 Viterbi 译码器将其输入 FIFO 中的数据译码完成之后进入步骤 (8);

[0022] (8) 读取当前 Viterbi 译码器的输出 FIFO 中的数据并输出,当该 Viterbi 译码器的输出 FIFO 读空之后,读取下一个 Viterbi 译码器的输出 FIFO 中的数据并输出。

[0023] 所述步骤 (1) 中插入虚拟比特按照如下步骤进行:

[0024] (a) 根据虚拟比特参数判断是否需要插入虚拟比特,若需要插入,则将输入数据按照打孔图样插入虚拟比特并写入数据输出缓存中,同时,将对应打孔图样的虚拟比特指示信号同步写入指示信号缓存中,之后进入步骤 (b);若不需要插入,则将输入数据直接输出;所述虚拟比特参数和打孔图样根据所使用的卷积码编码类型确定;

[0025] (b) 判断步骤 (a) 中所述数据输出缓存是否为空,若为空,则返回步骤 (a);若不

空,则同时读取数据输出缓存和指示信号缓存中的数据,并输出。

[0026] 本发明与现有技术相比的有益效果是:

[0027] (1) 本发明通过采用多级并行分段交错 Viterbi 译码方法,解决了卷积编码拆分为若干段独立码流之后,无法并行译码的问题,使得在 Viterbi 译码器的基础上可以将输入数据转换为并行数据进行并行 Viterbi 译码处理,大大加快了译码速率,使得现有技术当中 Viterbi 译码算法能够实现的最高译码速率有一个瓶颈的问题得到了妥善的解决。

[0028] (2) 本发明通过在虚拟比特插入时同步生成虚拟比特指示信号,不仅能够适应 CCSDS 标准规定的各种卷积编码打孔模式,而且可以方便的扩展为其它打孔模式。

[0029] (3) 本发明通过采用模块化设计,需要提高并行度时,直接增加 Viterbi 译码器的数量,修改相关参数即可进一步提高译码速率。

附图说明

[0030] 图 1 为 Viterbi 译码器组成图;

[0031] 图 2 是本发明虚拟比特插入流程图;

[0032] 图 3 是本发明高速分段交错 Viterbi 译码方法流程图。

具体实施方式

[0033] 下面结合附图对本发明的具体实施方式进行进一步的详细描述。

[0034] 卷积码是一种信道编码格式,它是应用最广泛的信道编码之一,维特比 (Viterbi) 译码算法是一种有效的卷积码译码算法。卷积码的主要用途是克服衰落信道中产生的随机错误,它还常常作为外码与 RS 码级联使用,以便获得更大的编码增益。伴随着空间通信技术的发展,数据传输码速率不断提高, Viterbi 译码速率也需要相应提高以适应发展需求。本发明主要解决 Viterbi 译码器实现中如何提高译码速率的问题,通过采用本发明提出的译码方法,可以大幅度提高译码速率,兼容各种卷积编码打孔模式,能够直接应用于高码速率遥感卫星地面接收系统。

[0035] 本发明一种高速并行分段交错维特比 (Viterbi) 译码方法步骤如图 3 所示:

[0036] (1) 将虚拟比特插入输入数据中,之后进入步骤 (2);所述输入数据为卷积编码器各支路输出交错排列后经 1:8 串并转换形成的 8bit 并行数据;数据发送端的卷积编码器输出数据格式为:

[0037] I 支路 C1C1C1C1;

[0038] Q 支路 C2C2C2C2 或 C2C2C2C2 (下划线表示取反,下同)。

[0039] I 路和 Q 路数据将送往调制器调制后发射;解调器将采用逆过程恢复 I 路和 Q 路数据。本译码方法的输入数据为 8bit 并行数据,该 8bit 并行数据指下述格式:

[0040] C1C2C1C2C1C2C1C2 或 C1C2C1C2C1C2C1C2 经过 1:8 串并转换后形成的各支路输出交错排列的并行数据。译码器采用软判决译码时,该并行数据需要相应的扩展为软判决量化位数。

[0041] (2) 复位第 i 个 Viterbi 译码器之后进入步骤 (3),所述 i 为大于等于 1 且小于等于 m 的自然数,i 初始化为 1,m 为 Viterbi 译码器的数量,且 $m \geq 2$;

[0042] (3) 将步骤 (1) 中得到的数据写入第 i 个 Viterbi 译码器的输入 FIFO 中,当写入

1 个字节后启动步骤 (7), 步骤 (7) 与步骤 (3) 并行, 意思是步骤 (3) 继续正常运行, 但是步骤 (7) 也同时运行, 两个是相互独立的进程, 同时运行; 当共写入了 $n-J$ 个字节之后, 进入步骤 (4); 所述 n 为每个 Viterbi 译码器单次译码处理的数据长度, 且 $n > 6k$, k 为卷积编码器的约束长度; 所述 J 为连续两个 Viterbi 译码器单次译码处理的数据的交错长度;

[0043] (4) 复位第 $i+1$ 个 Viterbi 译码器之后进入步骤 (5);

[0044] (5) 将步骤 (4) 中写入 $n-J$ 个字节之后的第一个字节写入第 i 个 Viterbi 译码器的输入 FIFO 和第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中之后启动步骤 (7), 步骤 (7) 与步骤 (5) 并行; 当共将步骤 (4) 中写入 $n-J$ 个字节之后的 J 个字节的数据同时写入第 i 个 Viterbi 译码器的输入 FIFO 和第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中之后, 再将写入 J 个字节之后的 $n-2J$ 个字节的数据写入第 $i+1$ 个 Viterbi 译码器的输入 FIFO 中, 之后进入步骤 (6);

[0045] (6) 判断 i 是否等于 m , 若等于, 则复位第 1 个 Viterbi 译码器并将步骤 (5) 中写入 $n-2J$ 个字节数据之后的 J 个字节的数据同时写入第 i 个 Viterbi 译码器的输入 FIFO 和第 1 个 Viterbi 译码器的输入 FIFO 中, 之后令 $i = 1$, 返回步骤 (3); 若不等于, 则将 $i+1$ 赋值给 i , 之后返回步骤 (4);

[0046] 步骤 (2) 至 (6) 即为分段交错 Viterbi 译码过程的数据分段交错过程, 它是为了解决 Viterbi 译码算法无法直接并行处理的问题。基本原理是先将输入数据先进行串并转换, 将时钟频率降低, 然后将输入数据分段交错后分配给多个译码器核, 从而实现并行处理。每个译码器中设置数据缓冲区, 将输入数据缓存后再译码。数据交错的目的在于保留一定长度的前一段数据, 进而保留相关信息, 否则下一段译码过程将出现错误。最小的相关长度应该是约束长度除以码率。对于卷积 (2, 1, 7) 码为:

[0047] $7/(1/2) = 14$

[0048] 在满足最小分段长度和交错深度的条件下, 如果将分段长度确定为 $n = 6\text{Byte} = 48\text{bits} > 42\text{bits}$ (译码截尾长度), 交错长度确定为 $J = 2\text{Byte} = 16\text{bits} > 14\text{bits}$ 。此时, 由于交错过程造成的数据冗余, 需要多个 Viterbi 译码器同时处理才能保证连续译码。假定使用 10 个 Viterbi 译码器, 当每个译码器译码速率为 100Mbit/s 的前提条件下, 整个 Viterbi 译码器总的译码速率将可达到:

[0049] $100 \times 10 = 1000\text{Mbit/s}$

[0050] 但由于数据交错造成了冗余, 因此实际译码速率将低于该值, 降低的比例将取决于交错长度与分段长度之比。在最后一个译码器完成数据输入后, 数据切回到第一个译码器, 此时该译码器应已经完成所缓存数据的译码, 并复位到初始状态, 以便接收下一段数据。每个译码器输入新的数据前均应复位, 以便回到初始状态。

[0051] 根据上述分析, 设分段长度为 n , 交错长度为 J , 所需的译码器数量为 m , 回溯深度为 t 。如果确定交错长度为 $J = 2\text{Byte}$, 输入到某一译码器的输入数据所需处理时间为: $8n+42$;

[0052] 该译码器下一次输入新数据的间隔为:

[0053] $(n-k)m-J$;

[0054] 则应满足下述关系:

[0055] $8n+t \leq (n-J)m-J$

[0056] n 和 m 均不应过大, 否则 FPGA 实现时硬件消耗会太大。n 和 m 均应为自然数, $m_{\min} = 8+1 = 9$ 。当取 $m = 10$ 时可得 $n = 32$ 。可以根据所需的译码速率和芯片容量选择合适的分组长度和并行译码器数量。

[0057] 为了在每个译码器输入新数据前使其可靠的复位, 可以多冗余 r 个时钟周期, 用于产生复位信号, 并确保译码器可靠工作。读出译码数据字宽为 8bit, 所以共需 $(n-2)/2$ 个时钟周期, 因此:

[0058] $8n+r+t+(n-1)/2 \leq (n-J)m$

[0059] 当 $m = 10$ 时, 取 $r = 14$ 可得 $n = 50$ 。

[0060] (7) 当前 Viterbi 译码器的输入 FIFO 非空时由该 Viterbi 译码器开始进行译码并将译码结果存入该 Viterbi 译码器对应的输出 FIFO 中, 当该 Viterbi 译码器将其输入 FIFO 中的数据译码完成之后进入步骤 (8);

[0061] 上述过程中使用 Viterbi 译码器进行译码为本领域的公知技术, 就是普通常用的 Viterbi 译码器以及译码方法。

[0062] (8) 读取当前 Viterbi 译码器的输出 FIFO 中的数据并输出, 当该 Viterbi 译码器的输出 FIFO 读空之后, 读取下一个 Viterbi 译码器的输出 FIFO 中的数据并输出。

[0063] 上述过程是分段交错并行 Viterbi 译码过程中的数据拼接过程, 各 Viterbi 译码器的输出为串行比特流, 数据拼接器根据各 Viterbi 译码器的数据输出 FIFO 的“满”信号驱动, 按照 8bit 字宽从各个 FIFO 依次读出译码后数据并输出。

[0064] 如图 2 所示, 插入虚拟比特按照如下步骤进行:

[0065] (a) 根据虚拟比特参数判断是否需要插入虚拟比特, 若需要插入, 则将输入数据按照打孔图样插入虚拟比特并写入数据输出缓存中, 同时, 将对应打孔图样的虚拟比特指示信号同步写入指示信号缓存中, 之后进入步骤 (b); 若不需要插入, 则将输入数据直接输出; 所述虚拟比特参数和打孔图样根据所使用的卷积码编码类型确定; 上面所述打孔图样为 CCSDS 标准规定的卷积编码所拥有的打孔模式, 如果卷积编码模式已知, 则打孔模式、打孔图样即为已知。

[0066] (b) 判断步骤 (a) 中所述数据输出缓存是否为空, 若为空, 则返回步骤 (a); 若不为空, 则同时读取数据输出缓存和指示信号缓存中的数据, 并输出

[0067] 经过大量仿真及工程测试, 使用该方法可使 Viterbi 译码速率达到 800Mbit/s, 优于目前文献中发表的同类型译码器的性能指标, 在需要时可以进一步通过增加并行度提高性能。

[0068] 本发明说明书中未作详细描述的内容属于本领域的公知技术。

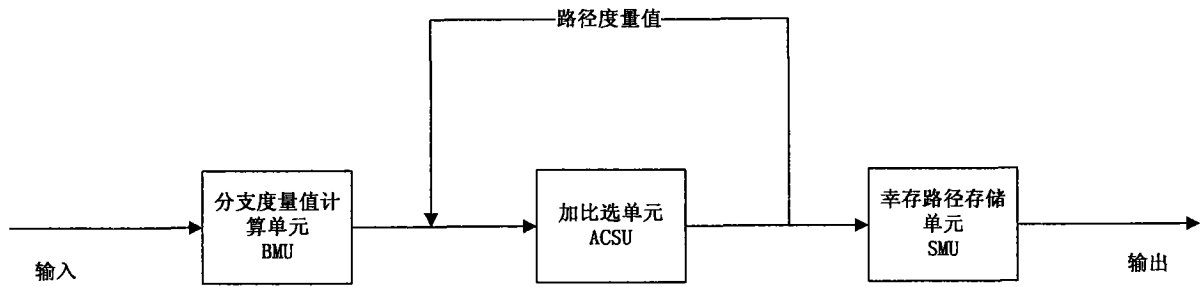


图 1

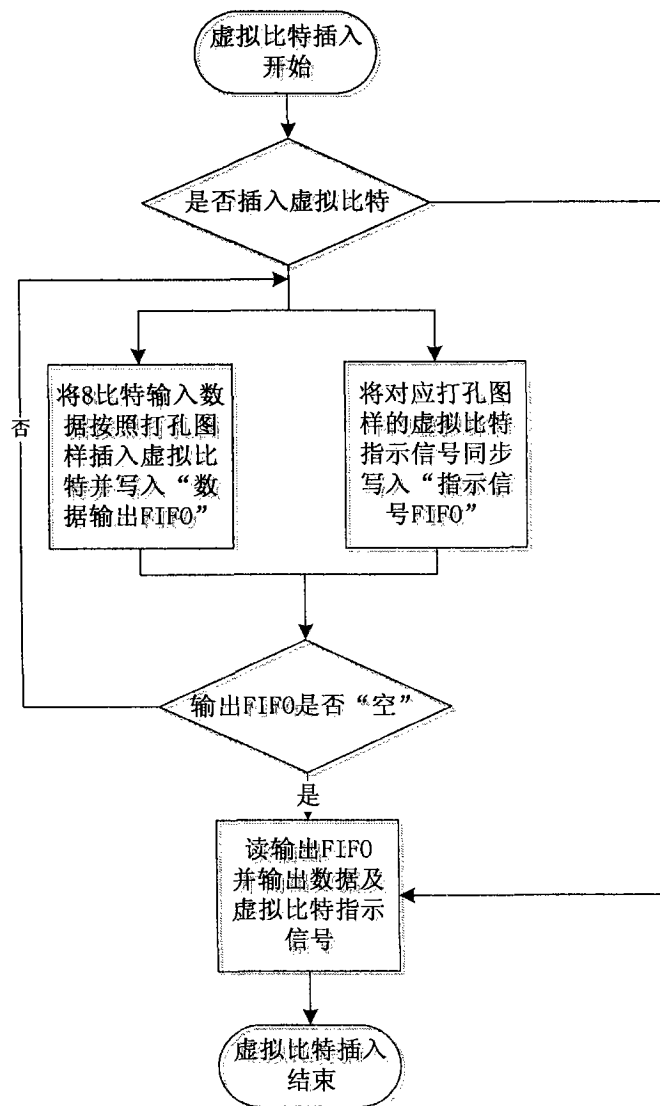


图 2

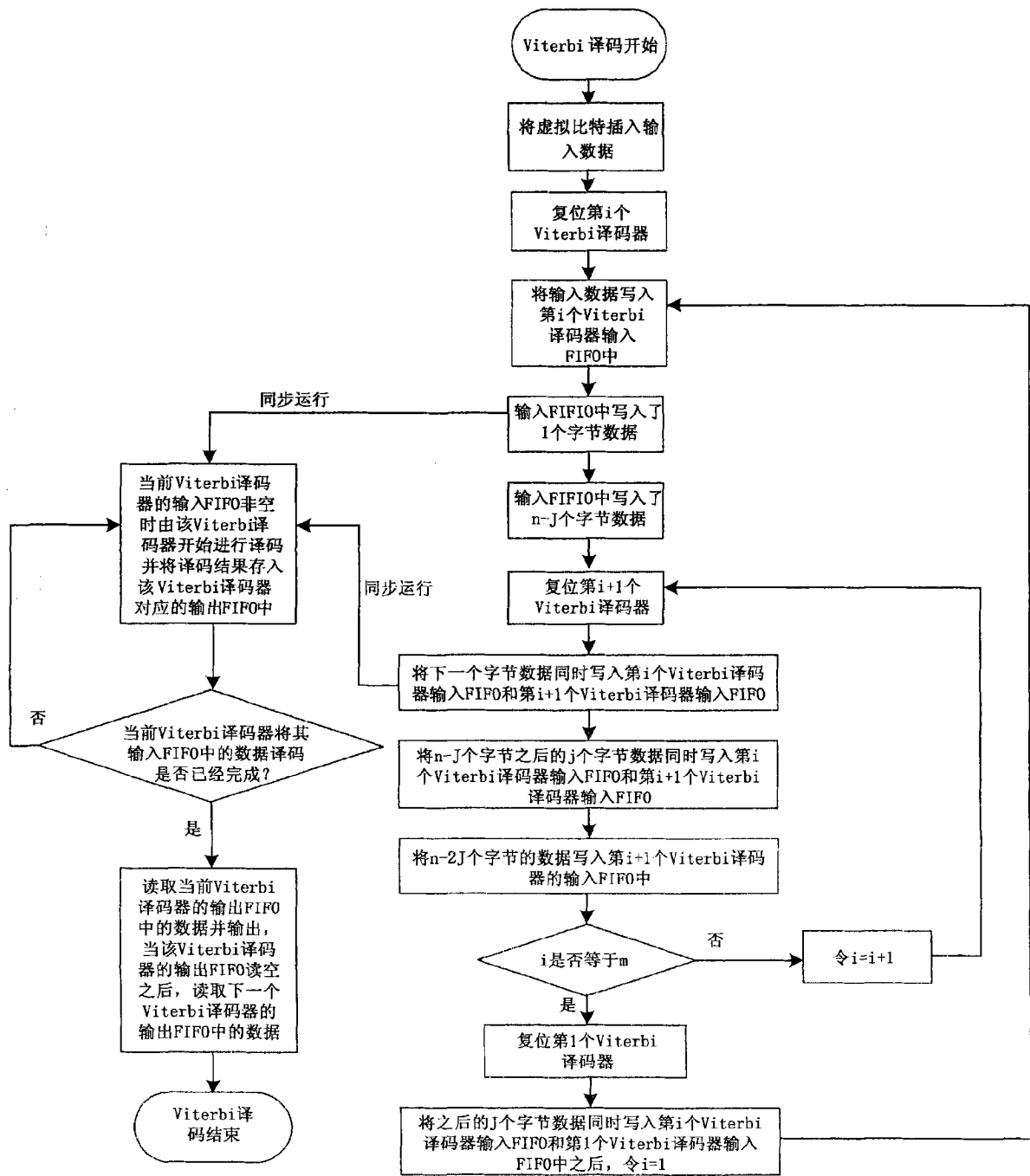


图 3