



(12) 发明专利申请

(10) 申请公布号 CN 103618556 A

(43) 申请公布日 2014. 03. 05

(21) 申请号 201310676642. 9

(22) 申请日 2013. 12. 11

(71) 申请人 北京理工大学

地址 100081 北京市海淀区中关村南大街 5 号

(72) 发明人 郑浩 李林涛 李祥明

(51) Int. Cl.

H03M 13/11 (2006. 01)

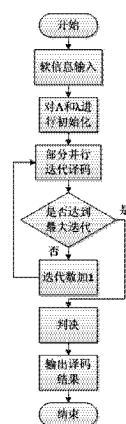
权利要求书2页 说明书6页 附图3页

(54) 发明名称

基于 RMP 调度的部分并行 QC-LDPC 译码方法

(57) 摘要

本发明涉及一种基于行消息传递(RMP)调度的部分并行 QC-LDPC 译码方法,属于通信技术领域。本发明在采用基于 RMP 调度的最小和译码算法的 QC-LDPC 译码器中实现部分并行的译码结构,在每次迭代译码的过程中,较最小和译码算法减少了近一半的迭代延时;针对 QC-LDPC 校验矩阵具有的准循环特点,采用了部分并行处理的译码结构,将校验矩阵进行分区,在分区内进行并行迭代译码,译码延时和每个分区内译码并行度呈线性反比关系,成倍地提高了译码器的吞吐量,并且保证了此并行方式与串行的 RMP 方式具有相同的性能,使 LDPC 译码器适应高速数据处理的要求。



1. 基于 RMP 调度的部分并行 QC-LDPC 译码方法,其特征在于:具体包括以下步骤:

步骤一,对行重为 a ,列重为 b 的 QC-LDPC 码的校验矩阵 $H(M, N)$ 进行分区,具体方法为:

(1) 寻找 QC-LDPC 码校验矩阵的最小循环子矩阵,并得到其大小为 $I \times I$, I 为常数;

(2) 在保证每个分区的列重为 1 的前提下,以每 J 行为一个分区将校验矩阵分为 K 个分区,其中 $I=nJ$, $M=KJ$, n 为整数,一般情况下 $n=1$,即 $I=J$;

(3) 确定分区内多路并行译码处理数 P ,有 $P1=J$, 1 为每个分区内每一路译码处理的行数;

步骤二,在分区的基础上,建立基于 RMP 调度的 QC-LDPC 码译码器,其组成包括:

变量节点软信息存储单元 RAM_{λ} ,用于对迭代译码过程中的初始化信息和变量节点软信息进行存储,其中包含 P 块 RAM 存储块 $M_{\lambda p}$;第 p 块 RAM 存储块 $M_{\lambda p}$,存储了每一个分区中的第 p 路译码迭代包含的行中的非零元素所对应的变量节点软信息;

校验节点软信息存储单元 RAM_{Λ} ,用于对迭代译码过程中更新的校验节点软信息进行存储,其中包含 P 块 RAM 存储块 $M_{\Lambda p}$;其中第 p 块 RAM 存储块 $M_{\Lambda p}$,存储了每一个分区中的第 p 路译码迭代包含的行中的非零元素所对应的校验节点软信息;

存储器地址产生模块 ADU,用于产生 QC-LDPC 译码器中所用的变量节点软信息存储单元和校验节点软信息存储单元;由 RAM_{λ} 存储器地址产生子模块和 RAM_{Λ} 存储器地址产生子模块组成,每个子模块由 P 个初始地址存储器和 P 个地址偏移计算器组成;每个子模块具有 P 个综合信号输出端口,其输出由初始地址存储器的来自 P 个初始地址存储器和计数器;ADU 有 $2P$ 个输出端口,分别与 RAM_{λ} 模块和 RAM_{Λ} 模块的 P 个读写端口的读写地址端口相连;

迭代译码模块 IDU,用于对迭代过程中的校验节点软信息和变量节点软信息进行并行更新运算,其中包含 P 个 CNU 计算模块;

译码判决模块 DJU,用于对变量节点软信息存储单元中即将输出的信息进行判决处理;

软信息交换模块 INU,用于在迭代译码时将来自 RAM_{λ} 模块中不同 RAM 存储块的数据送到相应的 CNU 计算模块,以及将相应的更新数据返回给 RAM 存储块;INU 有 $2P$ 输入端口和 $2P$ 个输出端口,其中 P 个输入端口和输出端口与 RAM_{λ} 模块中的 RAM 的输出端口及输入端口相连, P 个输入端口和输出端口与 P 个 CNU 计算模块中的 RAM 的输出端口及输入端口相连;通过 INU 模块综合信号,将 RAM_{λ} 模块中 P 个输出软信息分配到 P 个 CNU 模块中,并将 P 个 CNU 模块的输出软信息存储到 RAM_{λ} 模块中对应的 RAM 中;

译码流程控制模块 PCU,用于产生整个译码流程的控制信号,其中包括 INU 模块综合信号;

所述每块 RAM 存储块 $M_{\lambda p}$ 和 $M_{\Lambda p}$ 含有两个读写端口,其读写模式均是“先读后写”,每个读写端口均与第 p 块 CNU 计算模块相连,每个端口各负责一路数据的读写;

步骤三,对步骤二所建立的基于 RMP 调度的 QC-LDPC 码译码器的迭代译码器进行初始化:将接收到的一帧信道似然比较软信息信息,按照校验矩阵中的分区将信道信息存储到变量节点软信息存储单元 RAM_{λ} 的 P 块 RAM 存储块 $M_{\lambda p}$ 中,第 p 块存储块 $M_{\lambda p}$ 的数据与地址按照分区中的第 p 路对应的行中的非零元素所在的列进行对应,每个存储块的存储地址范围为 $0 \sim aJ-1$, a 为 QC-LDPC 码的行重, J 为每个分区第 p 路对应的行数;同时,将校验节

点软信息存储单元 RAM_Λ 中存储的数据全部初始化为 0,并将迭代次数 iter_time 初始化为 0 次;

步骤四,进行迭代译码运算:译码流程控制模块 PCU 和存储器地址产生模块 ADU 共同控制所述的迭代译码模块 IDU 和软信息存储模块进行运算更新;

步骤 4. a) 迭代译码模块 IDU 中的第 p 个 CNU 计算模块对第 k 个分区中的第 p 路并行信息进行迭代译码计算,模块输入的信息来自变量节点软信息存储单元 RAM_λ 和校验节点软信息存储单元 RAM_Λ,迭代译码的结果则保存到这两个存储单元相应的位置上;

其中,在更新该路第 i 行的信息时,与第 p 个 CNU 计算模块对应的变量节点软信息存储单元 RAM_λ 的 RAM 存储块,是其中的第 p 个 RAM 存储的信息按照存储器地址产生模块 ADU 给出的地址读出,RAM_λ 得到的地址可以按照如下计算得到:<第 k 个分区第 p 路译码初始化地址 addr_kp0>+<i-1>;

在更新该路第 i 行的信息时,与第 p 个 CNU 计算模块对应的校验节点软信息存储单元 RAM_Λ 的 RAM 存储块是其中的第 p 个 RAM 存储的信息按照存储器地址产生模块 ADU 给出的地址读出,RAM_Λ 得到的地址是按照存储块顺序的 a 个连续的地址,a 为 QC-LDPC 码的行重;

步骤 4. b) CNU 计算模块计算的结果通过软信息交换模块 INU 分别保存到变量节点软信息存储单元 RAM_λ 和校验节点软信息存储单元 RAM_Λ 对应的存储块中,写入的地址是由对应的读取地址进行 a 个时钟的延迟得到的;

步骤 4. c) 重复步骤 4. a) - 步骤 4. b),直到完成每个分区中的所有行的更新;

步骤 4. d),重复步骤 4. a) ~ 步骤 4. c),直到完成整个校验矩阵中所有分区的更新;

步骤五,重复步骤四,直到达到最大迭代次数,并将变量节点软信息存储单元 RAM_λ 中每个存储块中的数据按照存储的顺序读出进行判决,得到译码结果。

基于 RMP 调度的部分并行 QC-LDPC 译码方法

技术领域

[0001] 本发明涉及一种基于行消息传递(RMP)调度的部分并行 QC-LDPC 译码方法,属于通信技术领域。

背景技术

[0002] 低密度奇偶校验码(LDPC)是一种接近香农限的编码,其译码复杂度低、结构灵活,被广泛应用于现代通信系统,已经被多个通信与广播标准采纳。如数字卫星电视(DVB-S2)、无线局域网(WLAN)以及中国数字电视地面广播传输标准(DTMB)等。随着无线通信技术的发展,无线通信网络需具备更高速率的数据服务能力及综合实时多媒体业务,信道编码的快速译码成为迫切需求。

[0003] LDPC译码器多基于标准消息传递(SMP)的最小和算法(Min-Sum Algorithm)进行设计实现,该算法是一种对数似然比(Log-LLR)置信传播算法近似简化算法,虽然在译码性能上会有所损失,但硬件实现复杂度很低,译码过程中只存在数值比较和加减运算,适于工程实现。

[0004] 其译码器主要包括以下功能模块:变量节点软信息存储单元RAM_Q,与变量节点i对应存储;校验节点软信息存储单元RAM_R,与校验节点j对应存储;接收信息存储单元RAM_Y,与变量节点i对应存储;校验节点更新单元CNU,与校验节点j对应,完成校验节点更新的运算,并将结果返回给控制单元;变量节点更新单元VNU,与变量节点j对应,完成变量节点更新的运算,并将结果返回给控制单元;控制单元,用于产生各种控制信号控制译码器的读写地址,协调各个单元的工作。

[0005] 译码过程中,译码器接收到的软信息首先储存到接收信息存储单元RAM_Y中,随后开始初始化的过程:通过接收信息存储单元RAM_Y中的软信息对变量节点软信息存储单元RAM_Q进行更新;然后,进行校验节点更新:将变量节点软信息存储单元RAM_Q中的数据按照控制单元产生的读取地址读入校验节点更新单元CNU,完成运算,按照控制单元提供的存储地址将运算结果储存到校验节点软信息存储单元RAM_R中;接着,进行变量节点更新:将校验节点软信息存储单元RAM_R中的数据按照控制单元产生的读取地址读入变量节点更新单元VNU,完成运算,检测是否达到最大迭代次数,如果是则进行判决并将结果输出,否则按照控制单元提供的存储地址将运算结果储存到变量节点软信息存储单元RAM_Q中,并继续进行校验节点更新直到达到最大迭代次数。

[0006] 2005年Radosavljevic(“Optimized message passing schedules for LDPC decoding”, Signals, Systems and Computers, 2005:591-595)提出了一种基于行消息传递(RMP)的译码算法。

[0007] 上述译码过程的译码器主要包括以下功能模块:变量节点软信息存储单元RAM_N,与变量节点n对应存储;校验节点软信息存储单元RAM_M,与校验节点m对应存储;迭代译码更新单元ICU,完成迭代译码的运算,并将结果返回给控制单元;控制单元,用于产生各种控制信号控制译码器的读写地址,协调各个单元的工作。

[0008] 译码过程中,译码器接收到的软信息开始初始化的过程:将接收到的软信息存储到变量节点软信息存储单元 RAM_N 中,同时将校验节点软信息存储单元 RAM_M 进行初始化置零;然后,进行迭代更新:将变量节点软信息存储单元 RAM_N 和校验节点软信息存储单元 RAM_M 中的数据按照控制单元产生的读取地址读入迭代译码更新单元 ICU,完成运算,按照控制单元提供的存储地址将运算结果储存在校验节点软信息存储单元 RAM_N 和校验节点软信息存储单元 RAM_M 中;接着,检测译码迭代是否达到最大迭代次数,如果是则进行判决并将结果输出,否则继续进行校验节点更新直到达到最大迭代次数。

[0009] 但是,上述的两种算法设计的译码器都存在一些问题:

[0010] 1. 基于 SMP 的最小和算法每次更新都是用的上一次迭代之后的结果,每个节点更新的结果无法马上传递给其他节点,需要等到下一次迭代,导致算法的收敛速度慢。而且,每次迭代需要分别进行校验节点更新和变量节点更新,译码器每次迭代都要对存储器进行两次读取和写入操作。这些导致了译码需要更大的译码时延。

[0011] 2. 基于 RMP 的最小和译码算法虽然能解决一些基于 SMP 的最小和算法的问题,但是,从上面对基于 RMP 的最小和译码算法的介绍中可以看出,这种基于 RMP 的最小和译码算法进行更新时,后更新的行会使用到先更新的行中更新的数据,所以只能进行串行的更新,无法采用并行译码方式提高译码延时。

发明内容

[0012] 本发明的目的是为了降低了 QC-LDPC 码译码器的译码延时和采用 QC-LDPC 码作为信道编码的通信系统的通信延时,提出一种基于 RMP 调度的部分并行 LDPC 译码方法,针对 QC-LDPC 码,对基于 RMP 调度的 LDPC 译码器的译码处理结构进行了优化。

[0013] 本发明在采用基于 RMP 调度的最小和译码算法的 QC-LDPC 译码器中实现部分并行的译码结构,具体通过以下技术方案实现:

[0014] 步骤一,对行重为 a ,列重为 b 的 QC-LDPC 码的校验矩阵 $H(M, N)$ 进行分区,具体方法为:

[0015] (1) 寻找 QC-LDPC 码校验矩阵的最小循环子矩阵,并得到其大小为 $I \times I$, I 为常数;

[0016] (2) 在保证每个分区的列重为 1 的前提下,以每 J 行为一个分区将校验矩阵分为 K 个分区,其中 $I=nJ$, $M=KJ$, n 为整数,一般情况下 $n=1$,即 $I=J$;

[0017] (3) 确定分区内多路并行译码处理数 P ,有 $P1=J$, 1 为每个分区内每一路译码处理的行数。

[0018] 步骤二,在分区的基础上,建立基于 RMP 调度的 QC-LDPC 码译码器,其组成包括:

[0019] 变量节点软信息存储单元 RAM_{λ} ,用于对迭代译码过程中的初始化信息和变量节点软信息进行存储,其中包含 P 块 RAM 存储块 $M_{\lambda p}$;第 p 块 RAM 存储块 $M_{\lambda p}$,存储了每一个分区中的第 p 路译码迭代包含的行中的非零元素所对应的变量节点软信息;

[0020] 校验节点软信息存储单元 RAM_{Λ} ,用于对迭代译码过程中更新的校验节点软信息进行存储,其中包含 P 块 RAM 存储块 $M_{\Lambda p}$;其中第 p 块 RAM 存储块 $M_{\Lambda p}$,存储了每一个分区中的第 p 路译码迭代包含的行中的非零元素所对应的校验节点软信息;

[0021] 存储器地址产生模块 ADU,用于产生 QC-LDPC 译码器中所用的变量节点软信息存

储单元和校验节点软信息存储单元；由 RAM_{λ} 存储器地址产生子模块和 RAM_{Λ} 存储器地址产生子模块组成，每个子模块由 P 个初始地址存储器和 P 个地址偏移计算器组成；每个子模块具有 P 个综合信号输出端口，其输出由初始地址存储器的来自 P 个初始地址存储器和计数器；ADU 有 $2P$ 个输出端口，分别与 RAM_{λ} 模块和 RAM_{Λ} 模块的 P 个读写端口的读写地址端口相连。

[0022] 迭代译码模块 IDU，用于对迭代过程中的校验节点软信息和变量节点软信息进行并行更新运算，其中包含 P 个 CNU 计算模块；

[0023] 译码判决模块 DJU，用于对变量节点软信息存储单元中即将输出的信息进行判决处理；

[0024] 软信息交换模块 INU，用于在迭代译码时将来自 RAM_{λ} 模块中不同 RAM 存储块的数据送到相应的 CNU 计算模块，以及将相应的更新数据返回给 RAM 存储块；INU 有 $2P$ 输入端口和 $2P$ 个输出端口，其中 P 个输入端口和输出端口与 RAM_{λ} 模块中的 RAM 的输出端口及输入端口相连， P 个输入端口和输出端口与 P 个 CNU 计算模块中的 RAM 的输出端口及输入端口相连；通过 INU 模块综合信号，将 RAM_{λ} 模块中 P 个输出软信息分配到 P 个 CNU 模块中，并将 P 个 CNU 模块的输出软信息存储到 RAM_{λ} 模块中对应的 RAM 中。

[0025] 译码流程控制模块 PCU，用于产生整个译码流程的控制信号，其中包括 INU 模块综合信号。

[0026] 所述每块 RAM 存储块 $M_{\lambda p}$ 和 $M_{\Lambda p}$ 含有两个读写端口，其读写模式均是“先读后写”，每个读写端口均与第 p 块 CNU 计算模块相连，每个端口各负责一路数据的读写；

[0027] 步骤三，对步骤二所建立的基于 RMP 调度的 QC-LDPC 码译码器的迭代译码器进行初始化：将接收到的一帧信道似然比软信息信息，按照校验矩阵中的分区将信道信息存储到变量节点软信息存储单元 RAM_{λ} 的 P 块 RAM 存储块 $M_{\lambda p}$ 中，第 p 块存储块 $M_{\lambda p}$ 的数据与地址按照分区中的第 p 路对应的行中的非零元素所在的列进行对应，每个存储块的存储地址范围为 $0 \sim aJ-1$ ， a 为 QC-LDPC 码的行重， J 为每个分区第 p 路对应的行数；同时，将校验节点软信息存储单元 RAM_{Λ} 中存储的数据全部初始化为 0，并将迭代次数 $iter_time$ 初始化为 0 次；

[0028] 步骤四，进行迭代译码运算：译码流程控制模块 PCU 和存储器地址产生模块 ADU 共同控制所述的迭代译码模块 IDU 和软信息存储模块进行运算更新；

[0029] 步骤 4. a) 迭代译码模块 IDU 中的第 p 个 CNU 计算模块对第 k 个分区中的第 p 路并行信息进行迭代译码计算，模块输入的信息来自变量节点软信息存储单元 RAM_{λ} 和校验节点软信息存储单元 RAM_{Λ} ，迭代译码的结果则保存到这两个存储单元相应的位置上；

[0030] 其中，在更新该路第 i 行的信息时，与第 p 个 CNU 计算模块对应的变量节点软信息存储单元 RAM_{λ} 的 RAM 存储块，是其中的第 p 个 RAM 存储的信息按照存储器地址产生模块 ADU 给出的地址读出， RAM_{λ} 得到的地址可以按照如下计算得到： $\langle$ 第 k 个分区第 p 路译码初始化地址 $addr_kp0 \rangle + \langle i-1 \rangle$ ；

[0031] 在更新该路第 i 行的信息时，与第 p 个 CNU 计算模块对应的校验节点软信息存储单元 RAM_{Λ} 的 RAM 存储块是其中的第 p 个 RAM 存储的信息按照存储器地址产生模块 ADU 给出的地址读出， RAM_{Λ} 得到的地址是按照存储块顺序的 a 个连续的地址， a 为 QC-LDPC 码的行重；

[0032] 步骤 4. b) CNU 计算模块计算的结果通过软信息交换模块 INU 分别保存到变量节点软信息存储单元 RAM_{λ} 和校验节点软信息存储单元 RAM_{Λ} 对应的存储块中, 写入的地址是由对应的读取地址进行 a 个时钟的延迟得到的;

[0033] 步骤 4. c) 重复步骤 4. a) - 步骤 4. b), 直到完成每个分区中的所有行的更新;

[0034] 步骤 4. d), 重复步骤 4. a) ~ 步骤 4. c), 直到完成整个校验矩阵中所有分区的更新;

[0035] 步骤五, 重复步骤四, 直到达到最大迭代次数, 并将变量节点软信息存储单元 RAM_{λ} 中每个存储块中的数据按照存储的顺序读出进行判决, 得到译码结果。

[0036] 有益效果

[0037] 与传统的译码器相比, 本发明设计的译码器采用了基于 RMP 的译码算法, 在每次迭代译码的过程中, 较最小和译码算法减少了近一半的迭代延时; 另外, 针对 QC-LDPC 校验矩阵具有的准循环特点, 采用了部分并行处理的译码结构, 将校验矩阵进行分区, 在分区内进行并行迭代译码, 译码延时和每个分区内译码并行度呈线性反比关系, 从而成倍地提高了译码器的吞吐量, 并且保证了这种并行方式与串行的 RMP 方式具有相同的性能, 使 LDPC 译码器适应高速数据处理的要求。应用本发明的译码器设计, 可使 LDPC 译码器适应更多高吞吐量的通信应用场合。

附图说明

[0038] 图 1 为本发明 LDP {Calderbank, 1999#38} C 译码方法流程图;

[0039] 图 2 为具体实施方式中采用的 LDPC 译码器的结构示意图;

[0040] 图 3 为具体实施方式中 LDPC 译码器的存储器地址控制模块说明图;

[0041] 图 4 为具体实施方式中的地址分配方法说明图。

具体实施方式

[0042] 下面结合附图和实施例对本发明做进一步的说明。

[0043] 参照图 2 本发明提供的基于 RMP 的部分并行 QC-LDPC 译码器主要分为 6 个部分, 分别为变量节点软信息存储单元 RAM_{λ} , 校验节点软信息存储单元 RAM_{Λ} , 迭代译码模块 IDU, 译码判决模块 DJU, 译码流程控制模块 PCU。其中, 变量节点软信息存储单元 RAM_{λ} , 用于对迭代译码过程中的初始化信息和变量节点软信息进行存储, 其中包含 P 块 RAM 存储块 $M_{\lambda p}$; 校验节点软信息存储单元 RAM_{Λ} , 用于对迭代译码过程中更新的校验节点软信息进行存储, 其中包含 P 块 RAM 存储块 $M_{\Lambda p}$; 存储器地址产生模块 ADU, 用于产生 QC-LDPC 译码器中所用的变量节点软信息存储单元和校验节点软信息存储单元; 迭代译码模块 IDU, 用于对迭代过程中的校验节点软信息和变量节点软信息进行并行更新运算, 其中包含 P 个 CNU 计算模块; 译码判决模块 DJU, 用于对变量节点软信息存储单元中即将输出的信息进行判决处理; 译码流程控制模块 PCU, 用于产生整个译码流程的控制信号。其中, RAM_{λ} 和 RAM_{Λ} 模块中第 p 块 RAM 存储块, 存储了每一个分区中的第 p 路译码迭代包含的行中的非零元素所对应的变量节点和校验节点软信息; 每块 RAM 存储块含有两个读写端口, 其读写模式均是“先读后写”, 每个读写端口均与第 p 块 CNU 计算模块相连, 每个端口各负责一路数据的读写; 存储器地址产生模块 ADU, 由 RAM_{λ} 存储器地址产生子模块和 RAM_{Λ} 存储器地址

产生子模块组成,每个子模块由P个初始地址存储器和P个地址偏移计算器组成;存储器地址产生模块ADU,有2P个输出端口,分别与RAM_λ模块和RAM_Λ模块的P个读写端口的读写地址端口相连。

[0044] 译码器的各个模块之间的连接关系如下:

[0045] RAM_λ模块中的每一块RAM都含有两个读写端口,分别负责迭代译码软信息的读取和写入,其模式都为“先读取后写入”模式,第p块RAM的“读”端口同时与INU模块的第p个输入端口、译码判决模块DJU输入端口相连,第p块RAM的“写”端口同时与INU模块的第p个输入端口、译码器的输入端口相连;RAM_Λ模块中的每一块RAM都含有两个读写端口,分别负责迭代译码软信息的读取和写入,其模式都为“先读取后写入”模式,第p块RAM的两个端口同时与第p个迭代译码计算单元CNU_p的输入输出端口相连;INU模块的其中P个输出端口和输入端口分别与P个迭代译码计算单元CNU的输入端口和输出端口相连,另外P个输出端口和输入端口分别与RAM_λ模块中的P块RAM的输入端口和输出端口相连;ADU模块的2p个输出端口分别与RAM_λ模块以及RAM_Λ模块中的p个RAM读写地址端口相连,负责读写地址的控制。

[0046] 参照图1,本发明提供的迭代译码的方法,其步骤如下:

[0047] 步骤1,初始化:将接收到的一帧信道似然比较信息信息,按照校验矩阵中的分区将信道信息存储到变量节点软信息存储单元RAM_λ的P块RAM存储块M_{λp}中,第p块存储块M_{λp}的数据与地址按照分区中的第p路对应的行中的非零元素所在的列进行对应,每个存储块的存储地址范围为0~aJ-1,a为QC-LDPC码的行重,J为每个分区第p路对应的行数;同时,将校验节点软信息存储单元RAM_Λ中存储的数据全部初始化为0,并将迭代次数iter_time初始化为0次;

[0048] 步骤2,迭代译码运算,迭代译码模块IDU对软信息存储模块进行运算更新:

[0049] 迭代译码模块IDU中的每一个计算单元CNU_p以行的顺序逐个更新与之相连的RAM_λ和RAM_Λ模块的第p块RAM中的信息,P个计算单元同时对校验矩阵一个分区中的P路进行并行译码;

[0050] 所述的迭代译码模块IDU中的每个计算模块CNU_p对每个RAM中的信息的更新分为三步:CNU_p首先分别从RAM_λ和RAM_Λ模块的第p块RAM中读取更新所需要的软信息;CNU_p再根据读取的软信息进行迭代译码更新计算;最后将更新计算得到的外信息写回到RAM_λ和RAM_Λ模块的第p块RAM中;

[0051] 所述的迭代译码模块IDU对RAM_λ和RAM_Λ模块的每块RAM进行信息更新,需要对每块RAM中的软信息进行读出和写入,其中每块RAM的数据读写采用一种循环的地址读写管理方法,参照图4,其地址分配方法如下:

[0052] a) 校验节点软信息存储模块RAM_Λ按照每个分区的行顺序将第p路并行处理的校验软信息保存到第p块RAM_p中,1≤p≤P(P是并行处理路数,图中P=4),每个RAM的地址位为0~(aJ-1);

[0053] b) 每个子单位矩阵按照列顺序分为P个部分,每个子矩阵的第p个部分对应的变量节点软信息保存到RAM_λ模块的第p块RAM_p中,1≤p≤P(P是并行处理路数,图中P=4),每块RAM的地址位为0~J-1;

[0054] c) 在上述的前提下,ADU模块通过每个子矩阵的初始地址进行偏移计算,得到变

量节点软信息存储模块中每一块 RAM 所存储的变量节点软信息对应的校验节点软信息,并将结果送到 INU 模块中,对从 P 块 RAM 得到的变量节点软信息送到相应的计算模块 CNU 中进行迭代译码更新。

[0055] 例如,如果一个由 64×64 的单位矩阵循环得到的循环矩阵,其首地址为 23,进行 4 路并行译码,RAM_Λ 模块中的每一个 RAM 存储 8 个校验节点软信息,地址为 0 ~ 15,类似地,RAM_λ 模块中的每一个 RAM 存储 8 个校验节点软信息,地址为 0 ~ 15;每块 RAM 中存储的软信息地址与计算模块之间的关系如下表所示:

[0056]

迭代译码模块 IDU	校验节点软信息存储模块 RAM_Λ	变量节点软信息存储单元 RAM_λ
CNU_1	RAM_1	RAM_2: 7~15
		RAM_3: 0~6
CNU_2	RAM_2	RAM_3: 7~15
		RAM_4: 0~6
CNU_3	RAM_3	RAM_4: 7~15:
		RAM_1: 0~6
CNU_4	RAM_4	RAM_1: 7~15
		RAM_2: 0~6

[0057] 步骤 3,重复步骤 2,直到完成这个分区中每一路的更新;

[0058] 步骤 4,重复步骤 2 和步骤 3,直到完成整个校验矩阵中所有分区的更新,并将迭代次数 iter_time 加 1;

[0059] 步骤 5,重复步骤 2 到步骤 4,直到迭代次数 iter_time 达到最大值 iter_MAX,并将变量节点软信息存储单元 RAM_λ 中每个存储块中的数据按照存储的顺序读出进行判决,得到译码结果。

[0060] 与采用 SMP 最小和译码算法的 LDPC 译码器相比,在迭代次数与并行数相同的情况下,本发明的译码器能减少近一半译码延时;与采用 RMP 的 LDPC 译码器相比,在迭代次数相同的条件下,本发明的译码器能在不影响译码性能的情况下将减少一半的译码延时。

[0061] 以上所述为本发明的较佳实施例而已,本发明不应该局限于该实施例和附图所公开的内容。凡是不脱离本发明所公开的精神下完成的等效或修改,都落入本发明保护的范围。

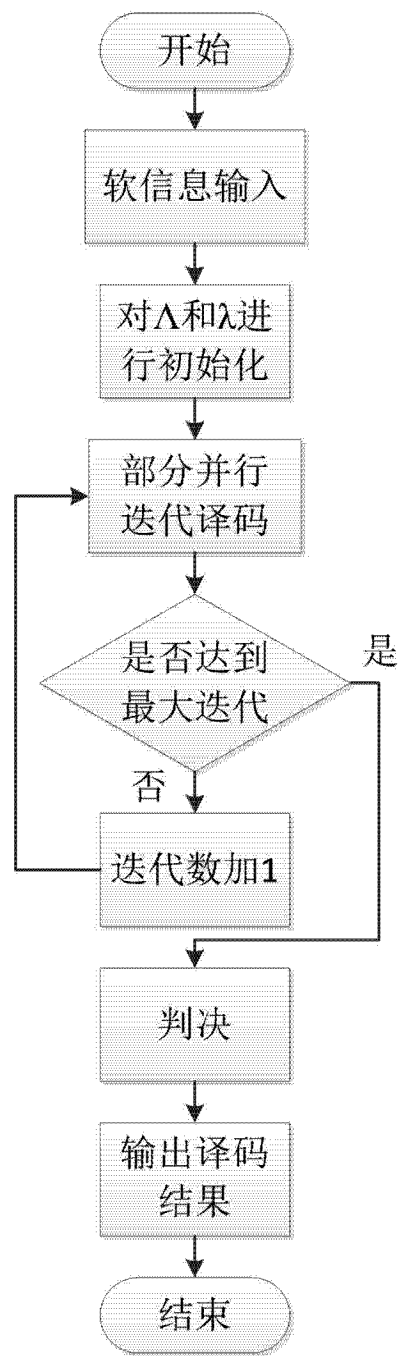


图 1

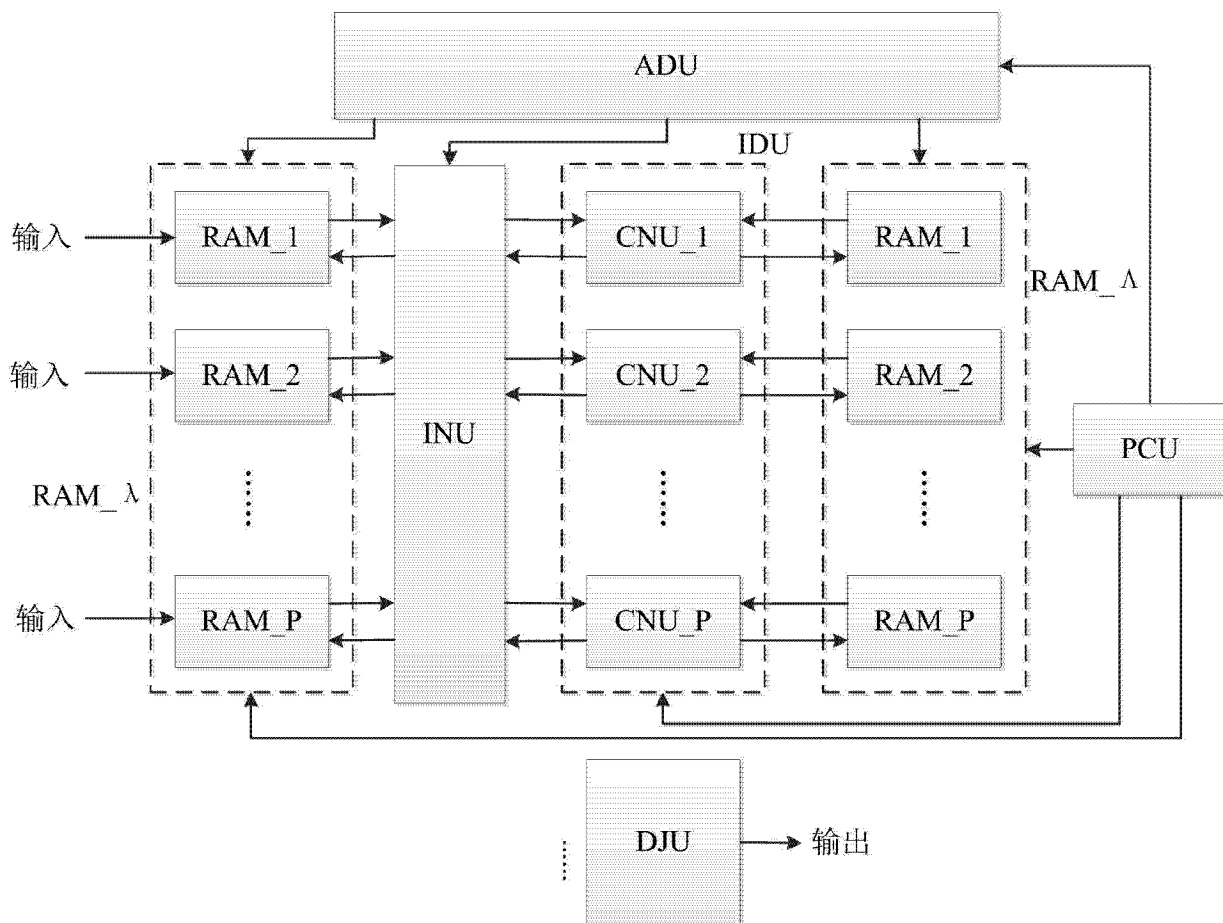


图 2

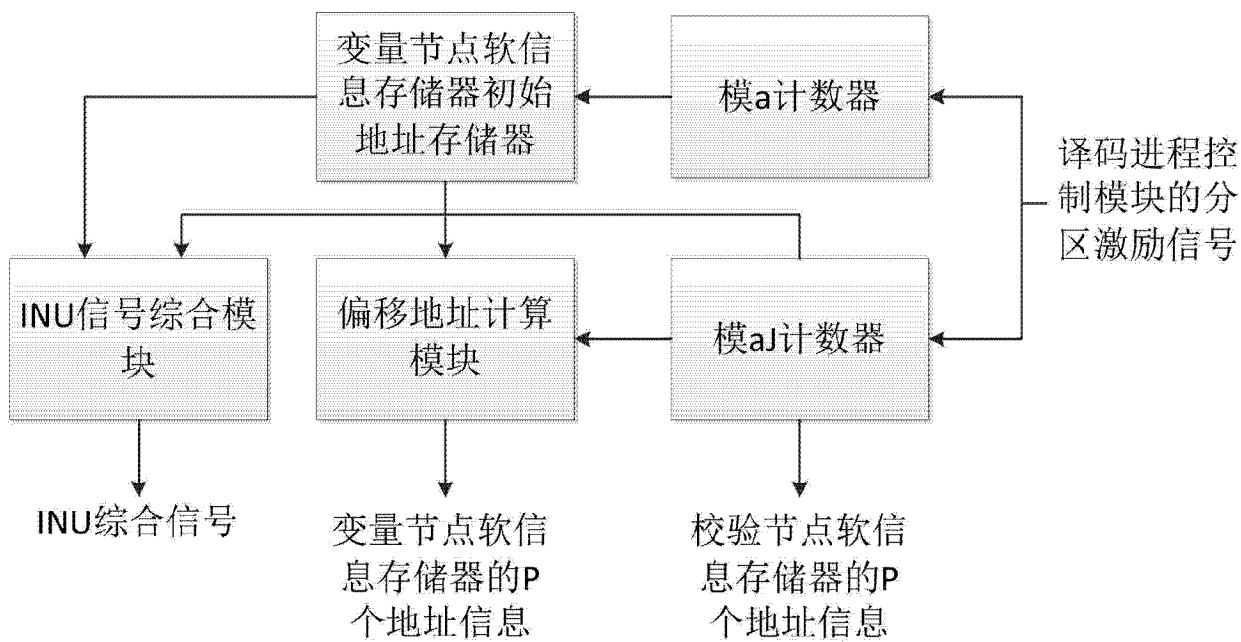


图 3

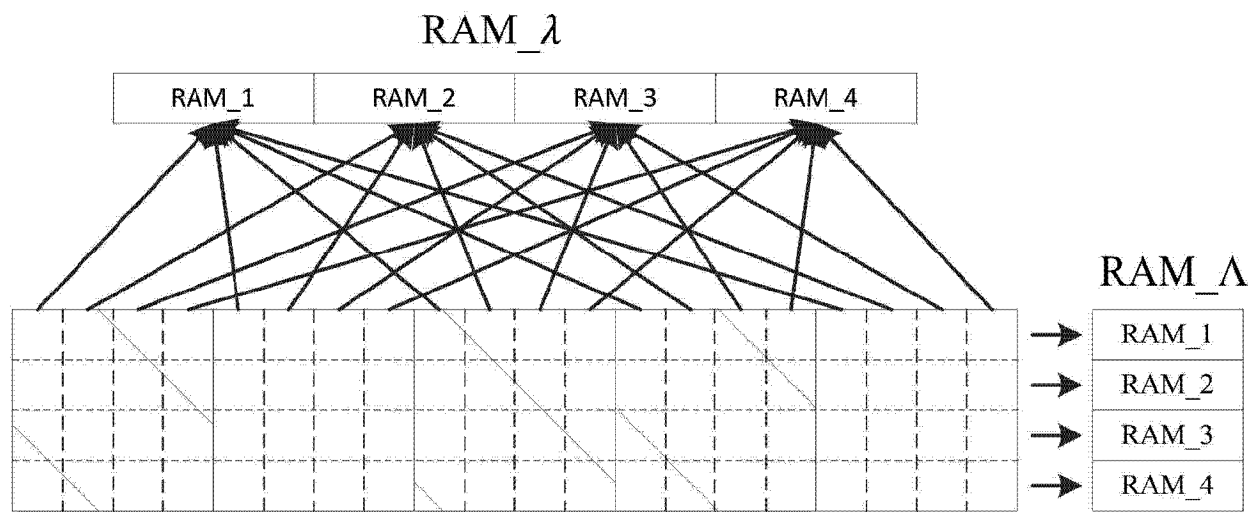


图 4