



(12)发明专利

(10)授权公告号 CN 105680877 B

(45)授权公告日 2019.06.28

(21)申请号 201410662824.5

(22)申请日 2014.11.19

(65)同一申请的已公布的文献号
申请公布号 CN 105680877 A

(43)申请公布日 2016.06.15

(73)专利权人 香港理工大学
地址 中国香港九龙红磡

(72)发明人 吕青 范剑峰 岑超荣 谭伟文
刘重明

(74)专利代理机构 深圳市顺天达专利商标代理
有限公司 44217

代理人 郭伟刚

(51)Int.Cl.
H03M 13/11(2006.01)

(56)对比文件

CN 102696176 A,2012.09.26,
CN 102035556 A,2011.04.27,
US 2010077277 A1,2010.03.25,
WO 2014122772 A1,2014.08.14,
Manabu HAGIWARA等.“Spatially Coupled
Quasi-Cyclic Quantum LDPC Codes”.《2011
IEEE International Symposium on
Information Theory Proceedings》.2011,

审查员 袁茹芳

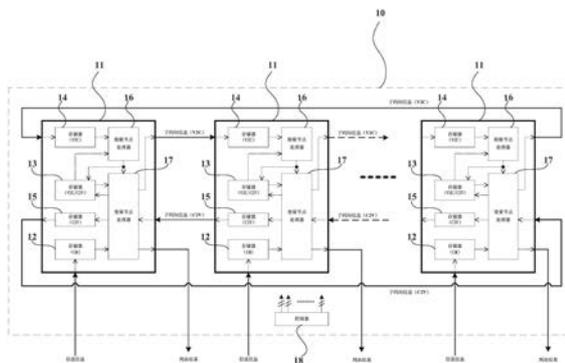
权利要求书2页 说明书11页 附图13页

(54)发明名称

一种CC-QC-LDPC码的构建方法及译码装置

(57)摘要

一种CC-QC-LDPC码的构建方法及译码装置,存储器包含4个分组12、13、14、15,分别存储指定的信息,子译码器11每次迭代包括与子码校验矩阵中子矩阵行数J相同数量的译码状态,每个译码状态更新一个层,每一层的z个校验节点被分为G个阶段处理,每个译码状态包括G个译码阶段,每个译码阶段内有z/G个校验节点并行处理并且有z/G个校验节点和相关联的变量节点被更新。实施本发明的有益效果是,LDPC码的长度得到了有效而又高效的扩展,译码准确度有了大幅度提高,降低了存储的需求量,译码装置的复杂度与误码率相近的其他编码相比较低并且有较高的吞吐率,整体性能优越。



1. 一种CC-QC-LDPC码的构建方法,其特征在于包含以下步骤:

将每一个子码的校验矩阵水平的划分为左边矩阵、中间矩阵以及右边矩阵,所述校验矩阵的每一个子矩阵都被完整划分到所述左边矩阵、中间矩阵或者右边矩阵中的一个;

连接所有的子码并进行循环混叠形成总码字,在混叠部分相邻的子码共同占有有一定数量的码位,前一个子码的右边矩阵与后一个子码的左边矩阵在总码字中占有相同的列,最后一个子码的右边矩阵与第一个子码的左边矩阵在总码字中占有相同的列。

2. 一种CC-QC-LDPC码的译码装置(10),用于CC-QC-LDPC码的译码,包括:用于对子码进行译码工作的子译码器(11)以及用于控制所述子译码器进行译码工作的全局控制器(18),所述子译码器包含层译码器和存储器,所述子码的校验矩阵由 $J \times L$ 个大小为 $z \times z$ 子矩阵构成,所述子矩阵为对单位矩阵进行列循环移位形成的循环矩阵,其特征在于:

所述层译码器包括变量节点处理器(17)和校验节点处理器(16),用于更新所对应子码的所有层的外部信息;

所述存储器用于存储关于每个变量节点的信道信息以及在译码过程中各个变量节点与各个校验节点之间传递的外部信息;

所述全局控制器(18)包括路由控制器和地址控制器,所述存储器和所述层译码器之间以及所述存储器和所述地址控制器之间通过数据选择器MUX连接;

所述存储器包含4个存储器组,分别记为第一存储器组(14)、第二存储器组(15)、第三存储器组(13)以及第四存储器组(12);

所述第一存储器组(14)用于存储相邻子码连接处由两个子码共占的变量节点传递给较后一个子码的校验节点的外部信息,并由两个不同的子码译码器进行读和写;

所述第二存储器组(15)用于存储相邻子码连接处由较后一个子码的校验节点传递给两个子码共占的变量节点的外部信息,并由两个不同的子码译码器进行读和写;

所述第三存储器组(13)用于存储所属子码内部校验节点与变量节点之间传递的外部信息,并仅由该子码对应的子码译码器进行读和写;

所述第四存储器组(12)用于存储所属子码内部变量节点的信道信息,由所属的子码译码器读取并在对一个码字完成译码后,由译码系统的输入端装载新的信道信息;

所述子译码器的一次译码过程包括预设次数的译码迭代,每次译码迭代包括与子码校验矩阵中子矩阵行数 J 相同数量的译码状态,每个译码状态更新一个层,每一层的 z 个校验节点被分为 G 个阶段处理,每个译码状态包括 G 个译码阶段,每个译码阶段内有 z/G 个校验节点并行处理并且有 z/G 个校验节点和相关联的变量节点被更新,所述的层是指所述子矩阵中的处于同一行的 z 个校验节点;

各个存储器组被划分为 z/G 个分组,每个分组包含深度为 G 的存储器单元,各个分组分别用于存储所述 z/G 个校验节点和相关联的变量节点被更新时产生的数据,所述深度为 G 的存储器单元分别用于存储所述 G 个译码阶段更新时产生的数据。

3. 如权利要求2所述的装置,其特征在于,每个所述的译码阶段包括四个时钟,每个时钟分别完成数据读取、校验节点更新、变量节点更新和数据写入, G 个外部信息组以流水线技术的方式处理,当读取到的数据为第 $J+1$ 层的第1组数据时开始新的译码状态。

4. 根据权利要求2所述的装置,其特征在于,所有的子译码器(11)同时工作,对CC-QC-LDPC码的所有子码并行处理。

5. 根据权利要求2所述的装置,其特征在于,所述的第一存储器组(14)对变量节点传递给校验节点的外部信息的存储采用两个存储器套组交替使用的方式,交替的存储有关一个子码奇数层和偶数层校验节点的外部信息。

6. 根据权利要求2所述的装置,其特征在于,所述的第三存储器组(13)动态地交替存储变量节点传递给校验节点和同一个校验节点传递给同一个变量节点的外部信息。

7. 根据权利要求2所述的装置,其特征在于,所述校验节点处理器(16)由一系列并行的校验节点处理单元(161)组成,每一个处理单元为一个并行路由的查找表LUT阵列。

8. 根据权利要求2所述的装置,其特征在于,所述的变量节点处理器(17)由一系列并行的变量节点处理单元(171)组成,每一个处理单元为一组加法器阵列。

9. 根据权利要求2所述的装置,其特征在于,所述校验节点处理器(16)按公式 α_{mn} 进行更新校验节点单元(161)传递给变量节点单元(171)的信息,

$$\alpha_{mn} = 2 \tanh^{-1} \left(\prod_{n' \in \mathcal{N}(m) \setminus n} \tanh \left(\frac{\beta_{mn'}}{2} \right) \right),$$

所述变量节点处理器(17)按公式 β_{mn} 更新变量节点单元(171)传递给校验节点单元(161)的信息,按公式 β_n 更新后验概率信息,

$$\beta_{mn} = \lambda_n + \sum_{m' \in \mathcal{M}(n) \setminus m} \alpha_{m'n},$$

$$\beta_n = \lambda_n + \sum_{m \in \mathcal{M}(n)} \alpha_{mn},$$

其中, α_{mn} 表示由校验节点m传递到变量节点n的外部信息, β_{mn} 表示由变量节点n传递到校验节点m的外部信息, λ_n 表示变量节点n的信道信息, $\mathcal{M}(n)$ 表示与变量节点n相连接的所有校验节点的集合, $\mathcal{M}(n) \setminus m$ 表示 $\mathcal{M}(n)$ 中除去校验节点m, $\mathcal{N}(m)$ 表示与校验节点m相连接的所有变量节点的集合, $\mathcal{N}(m) \setminus n$ 表示 $\mathcal{N}(m)$ 中除去变量节点n, β_n 表示变量节点n的后验概率信息。

10. 根据权利要求2所述的装置,其特征在于,在所述子译码器(11)的一次译码过程最后一次迭代中,变量节点处理器(17)根据当时更新的后验概率信息对所对应的码位进行判决:当 $\beta_n \geq 0$ 时,判决值为0,否则判决值为1,在使用补码表示时,判决结果可以通过输出 β_n 的符号位得到。

一种CC-QC-LDPC码的构建方法及译码装置

技术领域

[0001] 本发明涉及通讯领域中关于前向差分纠错技术领域,更具体地说,涉及一种CC-QC-LDPC码的构建方法及译码装置。

背景技术

[0002] 低密度奇偶校验(LDPC)码是一种线性分组码,由于这类编码具有接近信道容量的特性,其在无线通信、卫星通信等通信领域得到了广泛应用。

[0003] 准循环低密度奇偶校验码(QC-LDPC)是LDPC码中一个结构性较强的子类,其特点是译码器结构简单而同时译码效果并不低于任意结构的LDPC码,因而在LDPC译码装置的设计中被广泛使用。

[0004] LDPC码通常有两种表示方式。第一种表示是一个只含有元素“0”和“1”的稀疏矩阵H,称为校验矩阵;第二种表示是一个由节点与连线构成的二分图,称为Tanner图。图1和图2分别示出了同一个QC-LDPC码的以上两种表示方式。H中的每一行与每一列分别对应于Tanner图中的每一个校验节点和每一个变量节点。H中的“1”表示码字中变量的约束关系,在Tanner图中表现为对应的变量节点与校验节点被一条线相连接。

[0005] 在实际应用中,LDPC码的长度会受到解码电路复杂度的限制,从而使得其误码率(BER)相比于具有同等解码器复杂度的卷积码而言有一定的差距。另一方面,卷积码在应用中同样具有缺点,如解码电路对存储器需求较大等。

发明内容

[0006] 本发明要解决的技术问题在于,针对现有技术的上述在实际应用中,LDPC码的长度会受到解码电路复杂度的限制,从而使得其误码率相比于具有同等解码器复杂度的卷积码而言有一定的差距以及卷积码在应用解码电路对存储器需求较大,提供一种CC-QC-LDPC码构建方法及译码装置。

[0007] 本发明解决其技术问题,所采用的技术方案是:构造一种CC-QC-LDPC码方法,包含以下步骤:

[0008] 将每一个子码的校验矩阵水平的划分为左边矩阵、中间矩阵以及右边矩阵,所述校验矩阵的每一个子矩阵都被完整划分到所述左边矩阵、中间矩阵或者右边矩阵中的一个。连接所有的子码并进行循环混叠形成总码字,在混叠部分相邻的子码共同占有有一定数量的码位,前一个子码的右边矩阵与后一个子码的左边矩阵在总码字中占有相同的列,最后一个子码的右边矩阵与第一个子码的左边矩阵在总码字中占有相同的列。

[0009] 为了解决上述技术问题,本发明还提供了一种CC-QC-LDPC码译码装置,用于循环耦合准循环低密度奇偶校验编码的译码,包括:用于对子码进行译码工作的子译码器以及用于控制所述子译码器进行译码工作的全局控制器,所述子译码器包含层译码器和存储器,所述子码由 $J \times L$ 个大小为 $z \times z$ 的子矩阵构成,子矩阵为对单位矩阵进行列循环移位得到的循环矩阵其特征在于:

[0010] 层译码器包括变量节点处理器和校验节点处理器,用于更新所对应子码的所有层的外部信息。

[0011] 存储器用于存储关于每个变量节点的信道信息以及在译码过程中各个变量节点与各个校验节点之间传递的外部信息。

[0012] 全局控制器包括路由控制器和地址控制器,所述存储器和所述层译码器之间以及所述存储器以及所述地址控制器之间通过数据选择器MUX连接,所有的MUX以及存储器的地址都要在所述路由控制器控制下工作。

[0013] 存储器包含4个存储器组,第一存储器组用于存储相邻子码连接处由两个子码共占的变量节点传递给校验节点的外部信息,并由两个不同的子码译码器进行读和写。第二存储器组用于存储相邻子码连接处由校验节点传递给两个子码共占的变量节点的外部信息,并由两个不同的子码译码器进行读和写。第三存储器组用于存储所属子码内部校验节点与变量节点之间传递的外部信息,并仅由该子码对应的子码译码器进行读和写。第四存储器组用于存储所属子码内部变量节点的信道信息,由所属子码的译码器读取并在对一个码字完成译码后,由所述译码系统的输入端装载新的信道信息。

[0014] 子码译码器的一次译码过程包括预设次数的译码迭代,每次译码迭代包括与子码校验矩阵中子矩阵行数J相同数量的译码状态,每个译码状态更新一个层,每一层的z个校验节点被分为G个阶段处理,每个译码状态包括G个译码阶段,每个译码阶段内有z/G个校验节点并行处理并且有z/G个校验节点和相关联的变量节点被更新,所述的层是指所述子矩阵中的处于同一行的z个校验节点。

[0015] 各个存储器组被划分为z/G个分组,每个分组包含深度为G的存储器单元,所述各个分组分别用于存储所述z/G个校验节点和相关联的变量节点被更新时产生的数据,所述深度为G的存储器单元分别用于存储所述G个译码阶段更新时产生的数据。

[0016] 在本发明所述的译码装置中,每个所述的译码阶段包括四个时钟,每个时钟分别完成数据读取、校验节点更新、变量节点更新和数据写入。G个外部信息组将以流水线技术的方式处理,当读取的数据为第J+1层的第1组数据时开始新的译码状态。

[0017] 在本发明所述的译码装置中,所有的子译码器同时工作,对循环耦合准循环低密度奇偶校验编码的所有子码并行处理。

[0018] 在本发明所述的译码装置中,第一存储器组对变量节点传递给校验节点的外部信息的存储采用两个存储器套组交替使用的方式,交替的存储有关一个子码奇数层和偶数层校验节点的外部信息。

[0019] 在本发明所述的译码装置中,第三存储器组动态地交替存储变量节点传递给校验节点和同一个校验节点传递给同一个变量节点的外部信息。

[0020] 在本发明所述的译码装置中,校验节点处理器由一系列并行的校验节点处理单元组成,每一个处理单元为一个并行路由的查找表LUT阵列。

[0021] 在本发明所述的译码装置中,变量节点处理器由一系列并行的变量节点处理单元组成,每一个处理单元为一组加法器阵列。

[0022] 在本发明所述的译码装置中,所述校验节点处理器按公式 α_{mn} 进行更新校验节点单元传递给变量节点单元的信息,

$$[0023] \quad \alpha_{mn} = 2 \tanh^{-1} \left(\prod_{n' \in \mathcal{N}(m) \setminus n} \tanh \left(\frac{\beta_{mn'}}{2} \right) \right)$$

[0024] 所述变量节点处理器按公式下述 β_{mn} 更新变量节点传递给校验节点的信息,按公式 β_n 更新后验概率信息;

$$[0025] \quad \beta_{mn} = \lambda_n + \sum_{m' \in \mathcal{M}(n) \setminus m} \alpha_{m'n}$$

$$[0026] \quad \beta_n = \lambda_n + \sum_{m \in \mathcal{M}(n)} \alpha_{mn}$$

[0027] α_{mn} 表示由校验节点 m 传递到变量节点 n 的外部信息, β_{mn} 表示由变量节点 n 传递到校验节点 m 的外部信息, λ_n 表示变量节点 n 的信道信息, $\mathcal{M}(n)$ 表示与变量节点 n 相连接的所有校验节点的集合, $\mathcal{M}(n) \setminus m$ 表示 $\mathcal{M}(n)$ 中除去校验节点 m , $\mathcal{N}(m)$ 表示与校验节点 m 相连接的所有变量节点的集合, $\mathcal{N}(m) \setminus n$ 表示 $\mathcal{N}(m)$ 中除去变量节点 n , β_n 表示变量节点 n 的后验概率信息。

[0028] 在所述子译码器的一次译码过程最后一次迭代中,变量节点处理器根据当时更新的后验概率信息对所对应的码位进行判决:当 $\beta_n \geq 0$ 时,判决值为0,否则判决值为1,在使用补码表示时,判决结果可以通过输出 β_n 的符号位得到。

[0029] 实施本发明的一种CC-QC-LDPC码的构建方法及译码装置,具有以下有益效果:通过采用循环耦合的方式,LDPC码的长度得到了有效而又高效的扩展,译码准确度有了大幅度提高,通过将存储器划分为4个分组并分别存储不同的信息,降低了存储的需求量,译码装置的复杂度与误码率相近的其他编码相比较低并且有较高的吞吐率,整体性能优越,为LDPC分组码代替卷积码而应用于高质量通信技术提供了一种可能性。

附图说明

[0030] 下面将结合附图及实施例对本发明作进一步说明,附图中:

[0031] 图1为一个QC-LDPC码校验矩阵 H 的例子;

[0032] 图2为图1中校验矩阵对应的Tanner图;

[0033] 图3为本发明CC-QC-LDPC构建方法实施例一循环耦合码构建过程的Tanner图的结构示意图;

[0034] 图4为本发明CC-QC-LDPC译码装置实施例二的结构示意图;

[0035] 图5为本发明CC-QC-LDPC译码装置实施例二的子译码器并行结构示意图;

[0036] 图6为本发明CC-QC-LDPC译码装置实施例二的译码流程图;

[0037] 图7为本发明CC-QC-LDPC译码装置实施例三中深度为14的校验节点处理单元结构示意图;

[0038] 图8为本发明CC-QC-LDPC译码装置实施例三各RAM组所存储的信息与校验矩阵的关系示意图;

[0039] 图9为本发明CC-QC-LDPC译码装置实施例三中RAM组14的工作方式示意图;

[0040] 图10为本发明CC-QC-LDPC译码装置实施例三中校验矩阵及存储器分配方案的示意图;

[0041] 图11A为图10中CC-QC-LDPC码的译码流程第一个状态第一阶段示意图;

点和相关联的变量节点被更新。

[0056] 本发明提出的译码装置架构10由K个子译码模块11和一个控制器18组成,每一个子译码模块11处理一个固定的子码。变量节点传递给校验节点(V2C)的信息以及校验节点传递给变量节点(C2V)的信息分别由变量节点处理器VNP 17以及校验节点处理器CNP 16更新并存储于相应的存储器(RAM)中。根据所存储的数据内容不同,存储器被分成四个RAM组:只存储信道信息(CM)的存储器12,动态存储变量节点传递给校验节点信息和校验节点传递给变量节点信息的存储器13,只存储变量节点传递给校验节点信息的存储器14以及只存储校验节点传递给变量节点信息的存储器15。

[0057] 一个层译码器(包括VNP和CNP)负责更新其所对应子码20的所有层的外部信息,每一个译码状态更新一个层。按照编码结构的要求,存储器与层译码器(包括VNP和CNP)之间通过数据选择器(MUX)进行连接,所有的MUX以及RAM的地址都要在全局控制器18的控制下工作和选择。

[0058] 本实施例中的译码装置按照子码间并行、子码内分层和层内分组的流程译码。在对每一个码字译码的开始,所有的子码译码器同时装载新的信道信息并同时开始译码直到译码结束。一个子码译码器的一次译码过程包括I次译码迭代,每次迭代包括J个译码状态,而每个译码状态包括G个译码阶段。

[0059] 这样,在每个译码阶段内共有z/G个校验节点被并行处理。参考图5,其为CC-QC-LDPC译码装置实施例二的子译码器并行结构示意图,校验节点处理器16和变量节点处理器17分别包含z/G个校验节点处理单元(CNPU) 161和z/G个变量节点处理单元(VNPU) 171以并行工作。每个变量节点处理单元171由L-W个加法器组成。为保证信息读取不发生冲突,各存储器RAM组30相应地划分成了z/G个分组31,每个分组31包括若干个深度为G的RAM,分别用于存储所述G个译码阶段更新时产生的数据。

[0060] 一个译码阶段包括四个时钟,每个时钟分别完成数据读取,校验节点更新,变量节点更新和数据写入四个操作步骤。G个外部信息组将以流水线技术的方式处理,即第一个时钟读取第一组数据,第二个时钟更新第一组校验节点同时读取第二组数据,第三个时钟更新第一组变量节点和第二组的校验节点同时读取第三组的数据……直到第j(1≤j≤J)层第G组的数据写入存储器,第j+1层(j=J时则为第1层)的第1组开始读取数据并开始新的状态。整体电路的译码过程如图6所示,其为为本发明CC-QC-LDPC译码装置实施例二的译码流程图。

$$[0061] \quad \alpha_{mn} = 2 \tanh^{-1} \left(\prod_{n' \in \mathcal{N}(m) \setminus n} \tanh \left(\frac{\beta_{mn'}}{2} \right) \right) \quad (1)$$

$$[0062] \quad \beta_{mn} = \lambda_n + \sum_{m' \in \mathcal{M}(n) \setminus m} \alpha_{m'n} \quad (2)$$

$$[0063] \quad \beta_n = \lambda_n + \sum_{m \in \mathcal{M}(n)} \alpha_{mn} \quad (3)$$

[0064] 在等式(1)(2)和(3)中, α_{mn} 表示由校验节点m传递到变量节点n的外部信息, β_{mn} 表示由变量节点n传递到校验节点m的外部信息, λ_n 表示变量节点n的信道信息, $\mathcal{M}(n)$ 表示与变

量节点n相连接的所有校验节点的集合, $M(n) \setminus m$ 表示 $M(n)$ 中除去校验节点m, $N(m)$ 表示与校验节点m相连接的所有变量节点的集合, $N(m) \setminus n$ 表示 $N(m)$ 中除去变量节点n, β_n 表示变量节点n的后验概率(APP)信息。

[0065] 根据BP译码算法, 变量节点处理器17按照等式(2)(3)更新V2C信息和APP信息, 每个处理单元171用一系列加法器来实现。在译码的最后一次(即第I次)迭代中, 变量节点处理器17根据当时更新的后验概率信息对所对应的码位进行判决: 当 $\beta_n \geq 0$ 时, 判决值为0, 否则判决值为1。在使用补码表示时, 判决结果可以通过输出 β_n 的符号位得到。

$$[0066] \quad O(I_i, I_j) = Q \left\{ 2 \tanh^{-1} \left(\tanh \frac{I_i}{2} \tanh \frac{I_j}{2} \right) \right\} \quad (4)$$

[0067] 校验节点处理器16按照等式(1)更新C2V信息, 每个处理单元161用一系列查找表LUT来实现。为了使整体的电路复杂度最低, 校验节点处理器16采用了只有两个输入的查找表来完成只有两个参数情况下的等式(1), 即等式(4)。其中Q表示量化运算。因此, 等式(1)可以通过递归地使用等式(4)来完成。

[0068] 实施例三:

[0069] 下面以一个深度为 $d_c=14$ 的校验节点的处理单元为例来描述校验节点处理器单元161的设计方法。校验节点的深度, 即Tanner图中一个校验节点所连接的变量节点的数量。

[0070] 一个14输入14输出的校验节点处理单元161由图7所示的拓扑图表示。图7中的每一个节点代表一个查找表, 每一个箭头代表了电路的连接以及数据传输方向。I和O分别表示该校验节点处理单元161的每个输入和输出。这样的校验节点处理单元可以通过以下两个步骤完成:

[0071] 1. 前向步骤。平行地放置 d_c-2 个LUT以形成一个平衡二叉树。这个二叉树的根节点的输出没有实际用处因而不需要放置对应的LUT。图6的下半部分表示了这个(除去根节点的)平衡二叉树。

[0072] 2. 后向步骤。连接每一个节点与其兄弟节点的子节点, 这样从上至下形成了通向每一个输入(树叶)的路径(图6中只用虚线表示了与输出 O_i 相关的路径和LUT)。对于路径中的每一段, 都用一个LUT实现并且使对应同一条路径的LUT级联, 这样每条路径最后一段对应的LUT的输出即为更新的C2V信息。

[0073] 采用查找表的方式构建校验节点处理器既不会引起较大的计算误差又尽可能的节省了硬件资源。相比于已有的基于查找表的校验节点更新电路, 本发明中的结构在同样的硬件资源条件下, 大幅度缩小了电路的时延, 提高了所允许的工作时钟频率。

[0074] 存储器的分配与校验矩阵的划分相对应。如图8所示, 其为本发明实施例三各RAM组所存储的信息与校验矩阵的关系示意图。RAM组12存储当前子码校验矩阵中 H_m 和 H_r 部分变量节点的信道信息 $C_{m,m}$ 和 $C_{m,r}$; RAM组13存储当前子码校验矩阵中 H_m 和 H_r 部分变量节点与当前子码校验矩阵中的校验节点 C_c 之间传递的外部信息, 包括变量节点传递给校验节点的信息 $V_{m,2C_c}$ 和 $V_{r,2C_c}$ 以及校验节点传递给变量节点的信息 $C_c, 2V_m$ 和 $C_c, 2V_r$; RAM组14存储当前子码校验矩阵中 H_1 部分(同时也是前一个子码校验矩阵中 H_r 部分)变量节点传递给当前子码的校验节点 C_c 的外部信息 $V_1, 2C_c$; RAM组15存储当前子码的校验节点 C_c 传递给当前子码校验矩阵中 H_1 部分(同时也是前一个子码校验矩阵中 H_r 部分)变量节点的信息 $C_c, 2V_1$ 。

[0075] 每个RAM的G个存储单元对应于G个校验节点并按照 z/G 为间隔采样。如果一个RAM的第一个存储单元涉及所在层的第 i 个校验节点 C_i ($1 \leq i \leq z/G$)，那么随后的存储单元涉及的该层校验节点为 $C_{i+z/G}, C_{i+2z/G}, \dots, C_{i+(g-1)z/G}, \dots, C_{i+(G-1)z/G}$ ，其中 g ($1 \leq g \leq G$)表示校验节点的组号。不同的RAM组采用不同的存储方案，各自的RAM数量由子码校验矩阵20分别决定。

[0076] RAM组12存储所属子码20中属于 H_m 和 H_r 部分变量节点的信道信息，信息数量为 $(L-W)z$ 。所以在分组数为G的条件下，RAM组12包含 $(L-W)z/G$ 个RAM。

[0077] RAM组13存储所属子码20中属于 H_m 和 H_r 部分变量节点与校验节点之间的外部信息。在每一个译码状态中，只有一层中的校验节点及相关变量节点被更新，所以在所发明的架构中，只有这一层的变量节点传递给校验节点的外部信息以及其他层的校验节点传递给变量节点的外部信息需要存储。因此，RAM组13中的每个存储单元动态地存储同一组变量节点和校验节点之间的V2C和C2V信息，使得每一对相连接的变量节点和校验节点只占用一个存储单元。所以，RAM组13总共包含 $J(L-W)z/G$ 个RAM。

[0078] RAM组15存储所属子码20中校验节点传递给属于 H_1 部分变量节点的外部信息。与RAM组13相同，每个变量节点和校验节点的连接占用一个存储单元，所以RAM组15总共包含 JWz/G 个RAM。

[0079] RAM组14存储所属子码20中属于 H_1 部分变量节点传递给校验节点的外部信息。由于这部分V2C信息由前一个子码译码器11输出并写入到RAM组14，而写入地址与当前子码译码器11对RAM组14的读取地址冲突，所以设置两个RAM套组交替地存储所属子码20中的奇数层与偶数层的信息。

[0080] 如图9所示，其为本发明CC-QC-LDPC译码装置实施例三中RAM组14的工作方式示意图。假设第一个状态更新当前子码20的第一层， H_1 部分的V2C₁信息从套组1读出。同时下一个状态下需要读取的第二层信息V2C₂由前一个子码译码器11写入套组2中。这样到了第二个状态，也就是当前子码20的第二层外部信息更新的时候，V2C₂从套组2读出，同时第三层信息V2C₃写入到套组1中并替换掉V2C₁……以此类推。所以RAM组14总共包含 $2Wz/G$ 个RAM。

[0081] 总体来说，在整个译码装置中，深度为G的RAM的用量为：

- [0082] • $K(L-W)z/G$ 个RAM存储信道信息；
- [0083] • $KJ(L-W)z/G$ 个RAM动态地存储V2C和C2V信息；
- [0084] • $KJWz/G$ 个RAM存储V2C信息；
- [0085] • $2KWz/G$ 个RAM存储C2V信息。

[0086] 下面参考具体例子来进一步描述本实例中译码装置的工作原理。在这个例子中各参数为 $K=2, L=4, J=2, W=1, z=G=3$ 。如图10所示，其为本发明CC-QC-LDPC译码装置实施例三中校验矩阵及存储器分配方案的示意图。

[0087] 每一层的3个校验节点被分成三组，每组包括一个校验节点。对于子码a，第一层的第一组校验节点为 C_1 ，第二组校验节点为 C_2 ，第三组校验节点为 C_3 ；第二层第一组校验节点为 C_4 ，第二组校验节点为 C_5 ，第三组校验节点为 C_6 。对于子码b，第一层的第一组校验节点为 C_7 ，第二组校验节点为 C_8 ，第三组校验节点为 C_9 ；第二层第一组校验节点为 C_{10} ，第二组校验节点为 C_{11} ，第三组校验节点为 C_{12} 。每个子码译码器使用13个RAM存储信道信息和外部信息，总共使用26个深度为3的RAM。

[0088] RAM_a1的第一个存储位置用于存储变量节点 V_1 传递给校验节点 C_1 的外部信息，第二

个存储位置存储变量节点V₁传递给校验节点C₂的外部信息,第三个存储位置存储变量节点V₃传递给校验节点C₃的外部信息。RAM_{a2}的第一个存储位置用于存储变量节点V₂传递给校验节点C₄的外部信息,第二个存储位置存储变量节点V₃传递给校验节点C₅的外部信息,第三个存储位置存储变量节点V₁传递给校验节点C₆的外部信息。

[0089] RAM_{a3}的第一个存储位置用于存储校验节点C₁传递给变量节点V₁的外部信息,第二个存储位置用于存储校验节点C₂传递给变量节点V₂的外部信息,第三个存储位置用于存储校验节点C₃传递给变量节点V₃的外部信息。RAM_{a4}的第一个存储位置用于存储校验节点C₄传递给变量节点V₂的外部信息,第二个存储位置用于存储校验节点C₅传递给变量节点V₃的外部信息,第三个存储位置用于存储校验节点C₆传递给变量节点V₁的外部信息。

[0090] RAM_{a5}的第一个存储位置用于存储变量节点V₅与校验节点C₁之间的外部信息,第二个存储位置用于存储变量节点V₆与校验节点C₂之间的外部信息,第三个存储位置用于存储变量节点V₄与校验节点C₃之间的外部信息。RAM_{a6}的第一个存储位置用于存储变量节点V₄与校验节点C₄之间的外部信息,第二个存储位置用于存储变量节点V₅与校验节点C₅之间的外部信息,第三个存储位置用于存储变量节点V₆与校验节点C₆之间的外部信息。

[0091] RAM_{a7}的第一个存储位置用于存储变量节点V₉与校验节点C₁之间的外部信息,第二个存储位置用于存储变量节点V₇与校验节点C₂之间的外部信息,第三个存储位置用于存储变量节点V₈与校验节点C₃之间的外部信息。RAM_{a8}的第一个存储位置用于存储变量节点V₉与校验节点C₄之间的外部信息,第二个存储位置用于存储变量节点V₇与校验节点C₅之间的外部信息,第三个存储位置用于存储变量节点V₈与校验节点C₆之间的外部信息。

[0092] RAM_{a9}的第一个存储位置用于存储变量节点V₁₁与校验节点C₁之间的外部信息,第二个存储位置用于存储变量节点V₁₂与校验节点C₂之间的外部信息,第三个存储位置用于存储变量节点V₁₀与校验节点C₃之间的外部信息。RAM_{a10}的第一个存储位置用于存储变量节点V₁₂与校验节点C₄之间的外部信息,第二个存储位置用于存储变量节点V₁₀与校验节点C₅之间的外部信息,第三个存储位置用于存储变量节点V₁₁与校验节点C₆之间的外部信息。

[0093] RAM_{a11}的第一个存储位置用于存储变量节点V₄的信道信息,第二个存储位置用于存储变量节点V₅的信道信息,第三个存储位置用于存储变量节点V₆的信道信息。RAM_{a12}的第一个存储位置用于存储变量节点V₇的信道信息,第二个存储位置用于存储变量节点V₈的信道信息,第三个存储位置用于存储变量节点V₉的信道信息。RAM_{a13}的第一个存储位置用于存储变量节点V₁₀的信道信息,第二个存储位置用于存储变量节点V₁₁的信道信息,第三个存储位置用于存储变量节点V₁₂的信道信息。

[0094] RAM_{b1}的第一个存储位置用于存储变量节点V₁₀传递给校验节点C₇的外部信息,第二个存储位置存储变量节点V₁₁传递给校验节点C₈的外部信息,第三个存储位置存储变量节点V₁₂传递给校验节点C₉的外部信息。RAM_{b2}的第一个存储位置用于存储变量节点V₁₁传递给校验节点C₇的外部信息,第二个存储位置存储变量节点V₁₂传递给校验节点C₈的外部信息,第三个存储位置存储变量节点V₁₀传递给校验节点C₉的外部信息。

[0095] RAM_{b3}的第一个存储位置用于存储校验节点C₇传递给变量节点V₁₀的外部信息,第二个存储位置用于存储校验节点C₈传递给变量节点V₁₁的外部信息,第三个存储位置用于存储校验节点C₉传递给变量节点V₁₂的外部信息。RAM_{b4}的第一个存储位置用于存储校验节点C₁₀传递给变量节点V₁₁的外部信息,第二个存储位置用于存储校验节点C₁₁传递给变量节点

V₁₂的外部信息,第三个存储位置用于存储校验节点C₁₂传递给变量节点V₁₀的外部信息。

[0096] RAM_{b5}的第一个存储位置用于存储变量节点V₁₄与校验节点C₇之间的外部信息,第二个存储位置用于存储变量节点V₁₅与校验节点C₈之间的外部信息,第三个存储位置用于存储变量节点V₁₃与校验节点C₉之间的外部信息。RAM_{b6}的第一个存储位置用于存储变量节点V₁₃与校验节点C₁₀之间的外部信息,第二个存储位置用于存储变量节点V₁₄与校验节点C₁₁之间的外部信息,第三个存储位置用于存储变量节点V₁₅与校验节点C₁₂之间的外部信息。

[0097] RAM_{b7}的第一个存储位置用于存储变量节点V₁₈与校验节点C₁₀之间的外部信息,第二个存储位置用于存储变量节点V₁₆与校验节点C₁₁之间的外部信息,第三个存储位置用于存储变量节点V₁₇与校验节点C₁₂之间的外部信息。RAM_{b8}的第一个存储位置用于存储变量节点V₁₈与校验节点C₁₀之间的外部信息,第二个存储位置用于存储变量节点V₁₆与校验节点C₁₁之间的外部信息,第三个存储位置用于存储变量节点V₁₇与校验节点C₁₂之间的外部信息。

[0098] RAM_{b9}的第一个存储位置用于存储变量节点V₂与校验节点C₇之间的外部信息,第二个存储位置用于存储变量节点V₃与校验节点C₈之间的外部信息,第三个存储位置用于存储变量节点V₁与校验节点C₉之间的外部信息。RAM_{b10}的第一个存储位置用于存储变量节点V₃与校验节点C₁₀之间的外部信息,第二个存储位置用于存储变量节点V₁与校验节点C₁₁之间的外部信息,第三个存储位置用于存储变量节点V₂与校验节点C₁₂之间的外部信息。

[0099] RAM_{b11}的第一个存储位置用于存储变量节点V₁₃的信道信息,第二个存储位置用于存储变量节点V₁₄的信道信息,第三个存储位置用于存储变量节点V₁₅的信道信息。RAM_{b12}的第一个存储位置用于存储变量节点V₁₆的信道信息,第二个存储位置用于存储变量节点V₁₇的信道信息,第三个存储位置用于存储变量节点V₁₈的信道信息。RAM_{b13}的第一个存储位置用于存储变量节点V₁的信道信息,第二个存储位置用于存储变量节点V₂的信道信息,第三个存储位置用于存储变量节点V₃的信道信息。

[0100] 子码a从它的第一层开始更新,子码b从它的第二层开始更新。每次迭代分为两个状态,每个状态包括三个阶段。对于每一个子码20,每个阶段有一个校验节点和三个变量节点被更新,因此校验节点处理器16和变量节点处理器17都只有一个处理单元,其中变量节点处理单元171由三个加法器组成。

[0101] 图11A-11D为图10所示CC-QC-LDPC码的译码流程示意图。参考图11A,在每次迭代的开始,RAM_{a5}、RAM_{a7}、RAM_{a9}和RAM_{b6}、RAM_{b8}、RAM_{b10}存储变量节点传递给校验节点的外部信息;RAM_{a6}、RAM_{a8}、RAM_{a10}和RAM_{b5}、RAM_{b7}、RAM_{b9}存储校验节点传递给变量节点的外部信息。

[0102] 第一个状态的第一阶段,子码a的校验节点C₁被更新。变量节点V₁、V₅、V₉和V₁₁传递给C₁的外部信息分别从RAM_{a1}、RAM_{a5}、RAM_{a7}和RAM_{a9}的第一个存储单元中读出,并输入到校验节点处理器16以得到更新的校验节点C₁传递给变量节点V₁、V₅、V₉和V₁₁的外部信息。这些更新的C_{2V}信息分别写入到RAM_{a3}、RAM_{a5}、RAM_{a7}和RAM_{a9}的第一个存储单元中,并且其中的C_{12V5}、C_{12V9}和C_{12V11}也输入到变量节点处理器17中。

[0103] 紧接着,变量节点V₅、V₉和V₁₁被更新。校验节点传递给这些变量节点的外部信息C_{52V5}、C_{42V9}、C_{62V11}、C_{82V11}、C_{102V11}分别从RAM_{a6}的第二个存储单元、RAM_{a8}的第一个存储单元、RAM_{a10}的第三个存储单元、RAM_{b3}的第二个存储单元、RAM_{b4}的第一个存储单元中读出。同时这些变量节点的信道信息CM₅、CM₉、CM₁₁分别从RAM_{a11}的第二个存储单元、RAM_{a12}的第三个存储单元、RAM_{a13}的第二个存储单元中读出。在该阶段中更新的校验节点传递给变量节点

的信息与以上所述的校验节点传递给变量节点的信息以及各变量节点的信道信息在变量节点处理器17中进行加法运算以得到更新后的变量节点传递给校验节点的外部信息 V_52C_5 、 V_92C_4 、 $V_{11}2C_6$ 、 $V_{11}2C_8$ 并分别写入RAM_a6的第二个存储单元、RAM_a8的第一个存储单元、RAM_a10的第三个存储单元、RAM_b1的第二个存储单元。同时， V_5 、 V_9 和 V_{11} 的后验概率信息也被更新。

[0104] 在同一个阶段，子码b中的校验节点 C_{10} 同时被更新。变量节点 V_{11} 、 V_{13} 、 V_{18} 和 V_3 传递给 C_{10} 的外部信息分别从RAM_b2、RAM_b6、RAM_b8和RAM_b10的第一个存储单元中读出，并输入到校验节点处理器中16以得到更新的校验节点 C_{10} 传递给变量节点 V_{11} 、 V_{13} 、 V_{18} 和 V_3 的外部信息。这些更新的C2V信息分别写入到RAM_b4、RAM_b6、RAM_b8和RAM_b10的第一个存储单元中，并且其中的 $C_{10}2V_{13}$ 、 $C_{10}2V_{18}$ 和 $C_{10}2V_3$ 也输入到变量节点处理器17中。

[0105] 紧接着，变量节点 V_{13} 、 V_{18} 和 V_3 被更新。校验节点传递给这些变量节点的外部信息 C_92V_{13} 、 C_72V_{18} 、 C_82V_3 、 C_32V_3 、 C_52V_3 分别从RAM_b5的第三个存储单元、RAM_b7的第一个存储单元、RAM_b9的第二个存储单元、RAM_a3的第三个存储单元、RAM_a4的第二个存储单元中读出。同时这些变量节点的信道信息 CM_{13} 、 CM_{18} 、 CM_3 分别从RAM_b11的第一个存储单元、RAM_b12的第三个存储单元、RAM_b13的第三个存储单元中读出。在该阶段中更新的校验节点传递给变量节点的信息与以上所述的校验节点传递给变量节点的信息以及各变量节点的信道信息在变量节点处理器17中进行加法运算以得到更新后的变量节点传递给校验节点的外部信息 $V_{13}2C_9$ 、 $V_{18}2C_7$ 、 V_32C_8 、 V_32C_5 并分别写入RAM_b5的第三个存储单元、RAM_b7的第一个存储单元、RAM_b9的第二个存储单元、RAM_a2的第二个存储单元。同时， V_{13} 、 V_{18} 和 V_3 的后验概率信息也被更新。

[0106] 一次迭代中的其余5个阶段可以依此推出并在图11B、图11C和图11D中示出。在外部信息更新的过程中，对于RAM组13(编号5-10的RAM)的存储单元，每次写入的外部数据类型与读出的外部数据类型都是不同的，因而变量节点传递给校验节点的外部信息与校验节点传递给变量节点的外部信息是动态交换存储的。

[0107] 存储器与处理器之间以及存储器与控制器之间的连接可以以RAM_a5为例来描述，如图12所示，其为图10所示CC-QC-LDPC码的译码装置路由电路示意图。可以看到，在处理器的输入端，RAM_a5与RAM_a6是交替连接的，而在存储器的输入端，校验节点处理器16和变量节点处理器17是交替连接的。这些连接都要通过数据选择器MUX并受到路由控制器的调配。此例子中的MUX有两个输入和一个输出，输出的选择取决于状态控制信号 S_0 。在第一个状态下，路由控制器输出的选择信号为低电平($S_0=0$)，此时RAM_a5的地址信号与初始值为0的计数器相连接，因而其第一个存储单元(地址为0)被访问。如前所述，此时的RAM_a5输出 V_52C_1 并且写入 C_12V_5 ，所以它的数据端口应该连接校验节点处理器16的输出端口，并且校验节点处理器16的对应输入端口也应该连接到RAM_a5的输出端口上。在第二个状态的第一个时钟下，RAM_a5输出 C_32V_4 并且写入 V_42C_3 ，此时路由控制器输出的选择信号为高电平($S_0=1$)，所以它的第三个存储单元(地址为2)被访问，其输入和输出端口相应地与变量节点处理器17的输出和输入端口相连接。每切换一次状态，路由控制器就会通过状态控制信号 S_0 切换相应端口的连接。

[0108] 图13为本发明译码装置的误码率仿真曲线，图中示出了四个不同大小的CC-QC-LDPC码和一个译码器复杂度相近的LDPC卷积码(LDPCCC)的误码率与信噪比关系曲线(4比特量化)。图中四个CC-QC-LDPC码的参数为 $K=4$ 、 $J=4$ 、 $L=28$ 、 $W=4$ 、 $I=10$ 。LDPCCC码参数为 $z=512$ 、 $G=512$ 、 $I=18$ 。其中，最长的CC-QC-LDPC码($z=1024$ ，码长98304比特)的误码率曲

线逼近于LDPC的误码率曲线。针对于该码的译码装置在Altera Stratix IV可编程逻辑阵列(FPGA)上的实现,共使用了79514个组合式自适应查找表(Combinational ALUT),43801个寄存器(Register)和2359296比特的存储空间。在100兆赫兹的时钟频率下,该译码装置的信息吞吐率可以达到1.55Gbps(千兆比特每秒)。在信噪比为3.50dB的条件下,对 1.14×10^{16} 比特信息的试验中没有找到错误译码。

[0109] 本领域技术人员应当理解,虽然在本发明实例中,循环耦合码的子码都是准循环(QC)码,但本发明并不局限于此。在本发明的其他实例中,每个子码可以有任意的结构特点。相应地,译码装置的构架需要进行相应的调整。

[0110] 本领域技术人员应当理解,虽然在本发明实例中,CC-QC-LDPC码的每个子码是相同的,但本发明并不局限于此。在本发明的其他实例中,每个子码可以有不同的结构和尺寸。相应地,每个子译码器占用的资源及其之间的协作方式需要相应地调整和控制。

[0111] 本领域技术人员应当理解,虽然在本发明实例中,对每一个子码分配一个固定的子译码器并且对所有子码同时进行处理,然而本发明并不局限于此。在本发明的其他实例中,可以只分配一个子译码器对所有的子码按一定顺序处理;或者分配多于一个的子译码器同时对一组子码进行处理,不同组按顺序处理。这样译码装置的硬件复杂度将进一步降低,从而节省硬件资源。

[0112] 本领域技术人员应当理解,虽然在本发明实例中,每个子译码器将一个层译码器复用以处理一个子码的所有层,然而本发明并不局限于此。在本发明的其他实例中,可以对每一层分别分配一个专属的层译码器,可以节省大量的数据选择器并使控制器简化。

[0113] 本领域技术人员应当理解,虽然在本发明实例中,每一层外部信息的处理按照先更新校验节点然后更新变量节点的顺序,然而本发明并不局限于此。在本发明的其他实例中,变量节点处理器17可以放置于校验节点处理器16之前。相应地,存储器中只存储校验节点传递给变量节点的外部信息,所占用的存储空间保持不变。

[0114] 本领域技术人员应当理解,虽然在本发明实例中,每一层的校验节点分成G组并分阶段处理,但本发明并不局限于此。在本发明的其他实例中,如果每一层的变量节点与校验节点连接足够小,可以同时处理一层当中的所有校验节点,这时候存储器可以用寄存器取代RAM来实现,控制器只需要控制信号的连接而不再需要控制地址。通过这种方式,可以有效减少译码时间,提高译码装置的吞吐率。

[0115] 本领域技术人员应当理解,虽然在本发明实例中,变量节点的后验概率信息由变量节点处理器输出并连续更新,但本发明并不局限于此。在本发明的其他实例中,可以通过额外的电路在迭代结束的时候更新变量节点的后验概率信息并由此做出判决,这样可以降低变量节点处理器17的复杂度。

[0116] 上面结合附图对本发明的实施例进行了描述,但是本发明并不局限于上述的具体实施方式,上述的具体实施方式仅仅是示意性的,而不是限制性的,本领域的普通技术人员在本发明的启示下,在不脱离本发明宗旨和权利要求保护的范围情况下,还可做出很多形式,这些均在本发明的保护之内。

$$H = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right]$$

图1

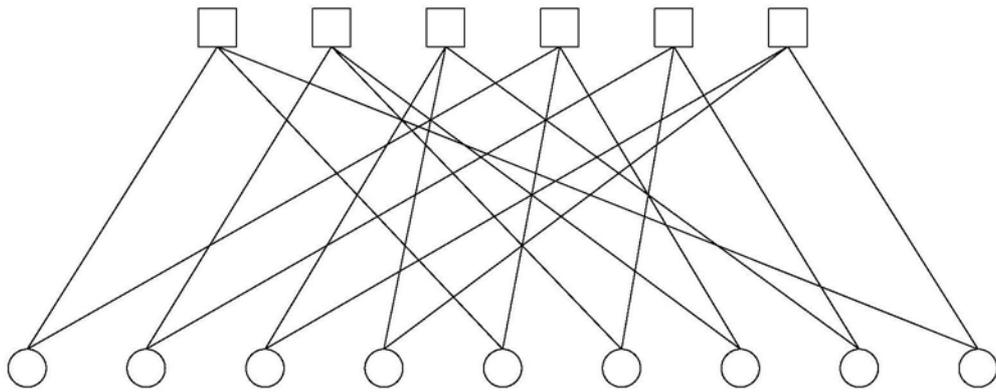


图2

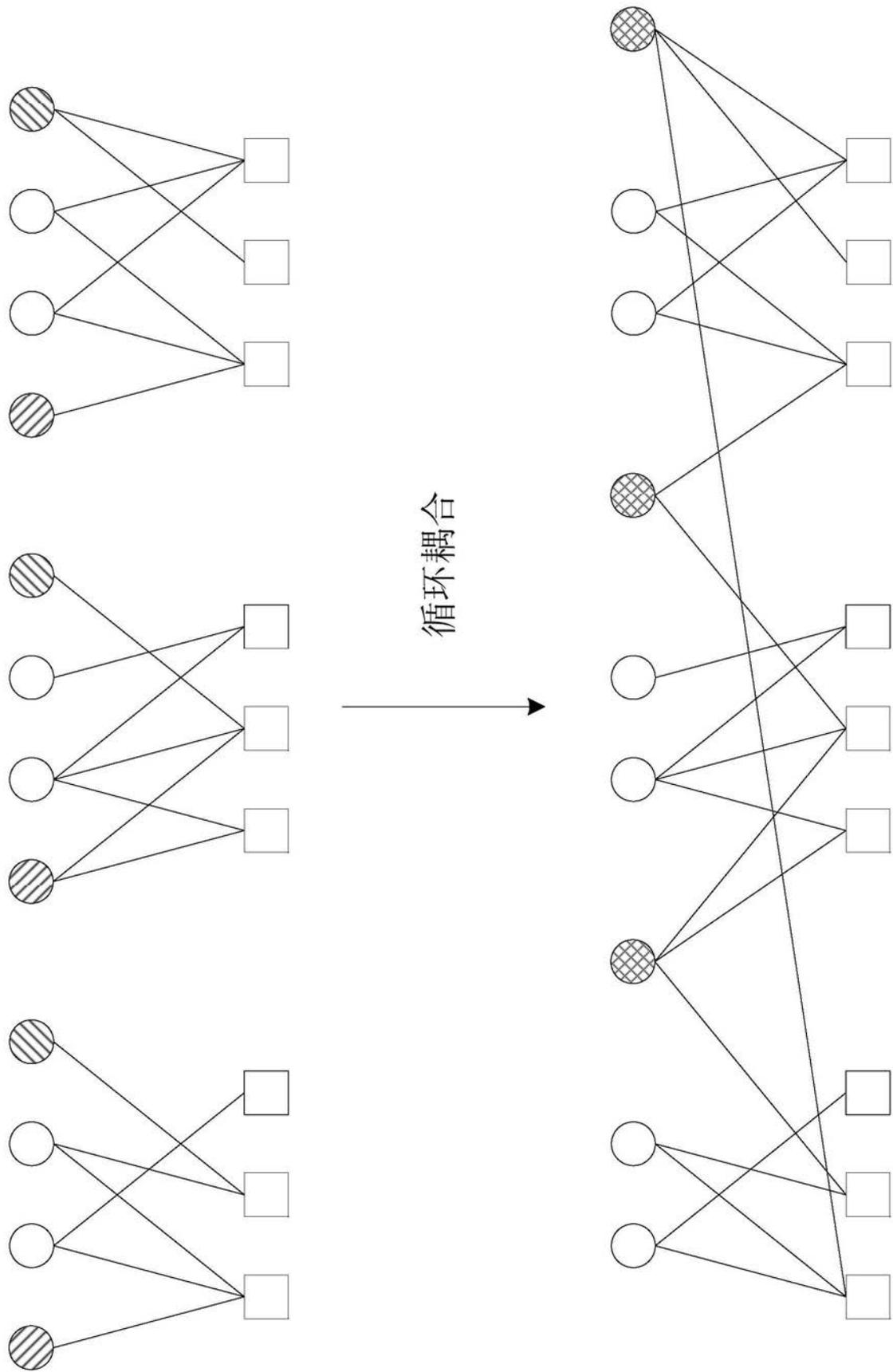


图3

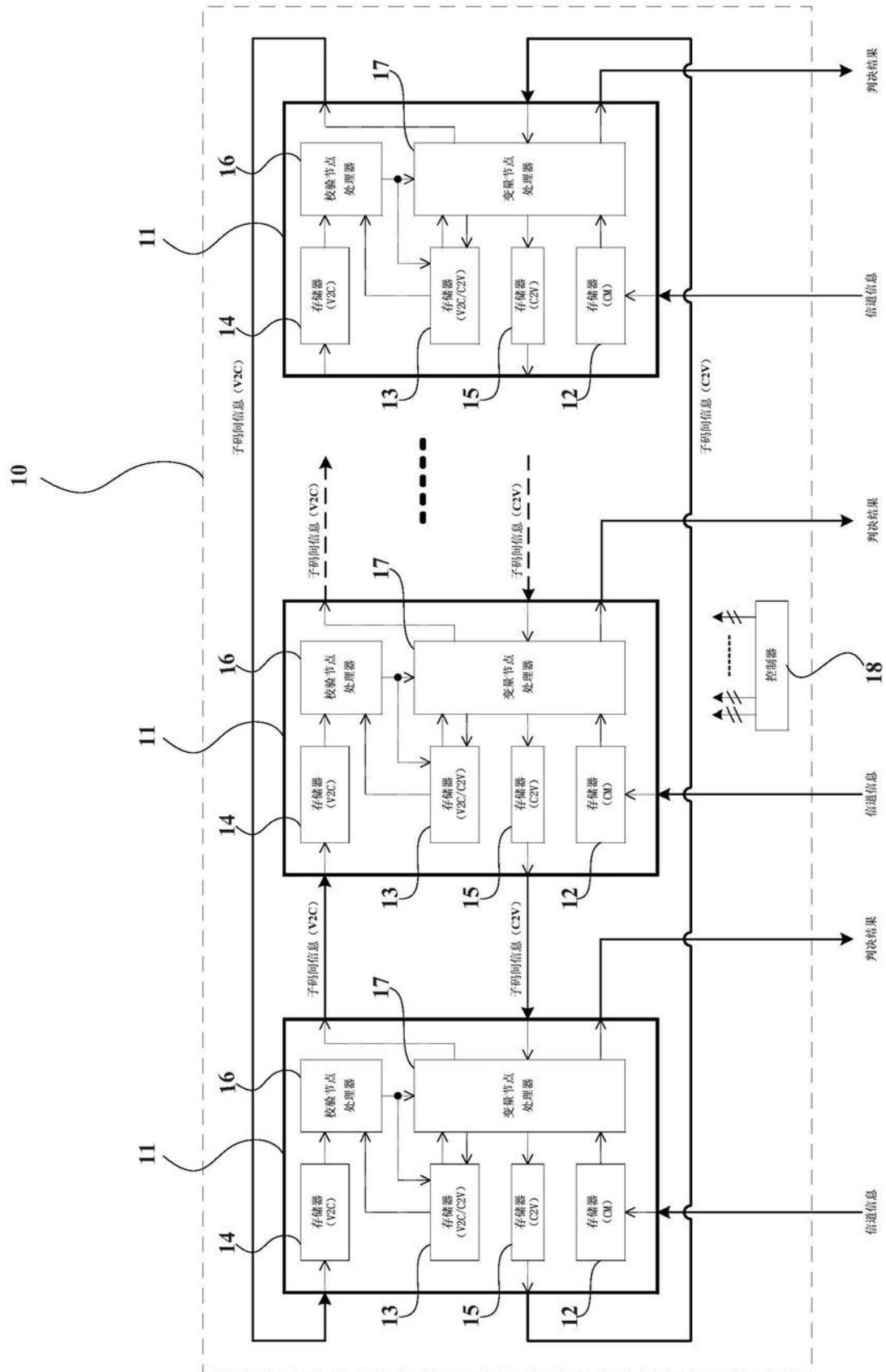


图4

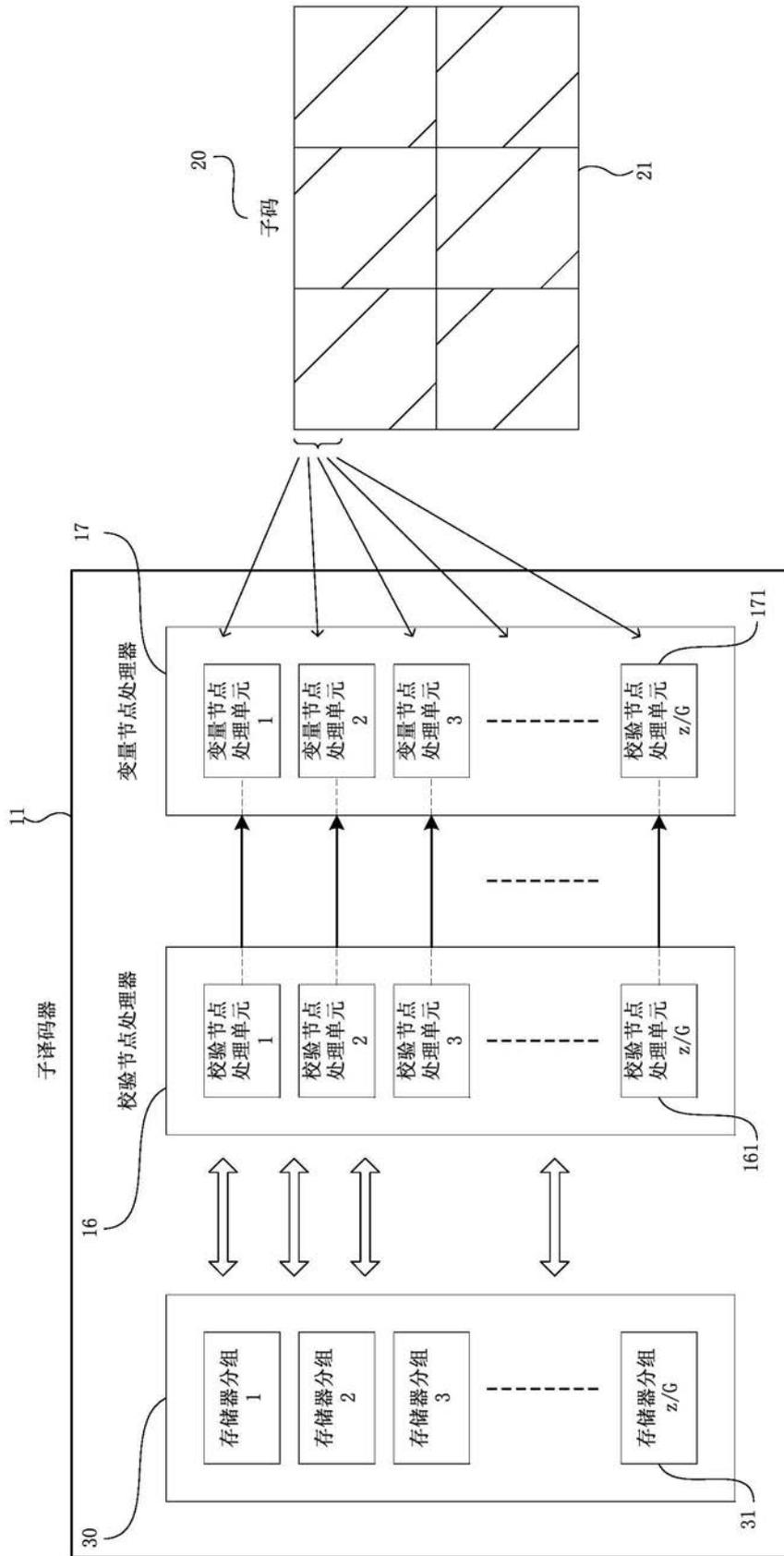


图5

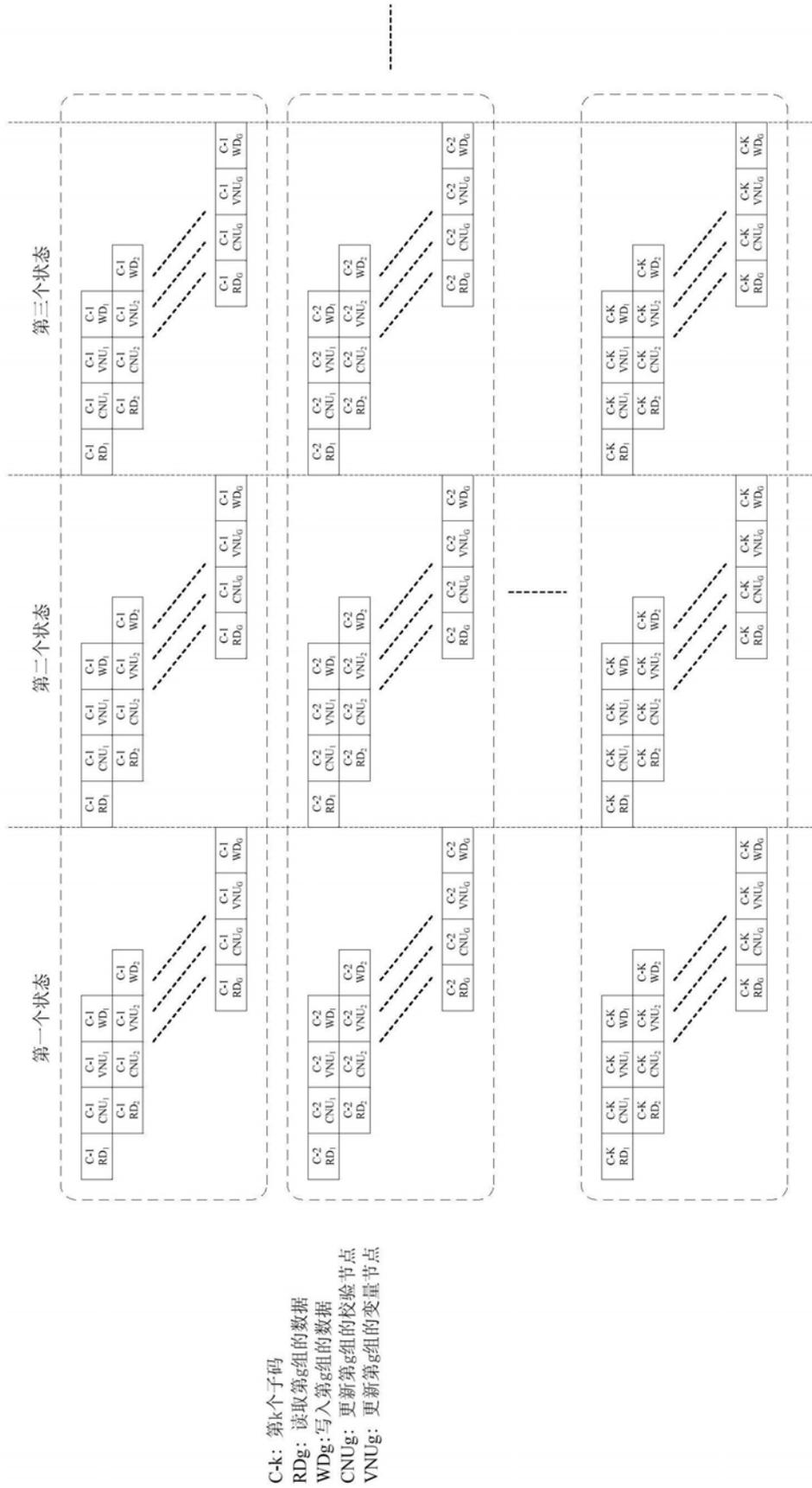


图6

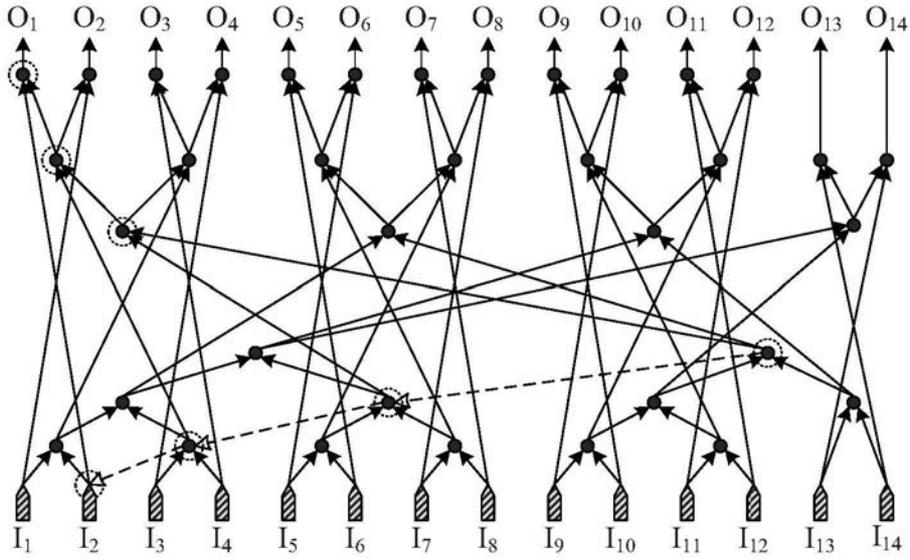


图7

C_p : 前一个子码的校验节点
 C_c : 当前子码的校验节点
 C_f : 后一个子码的校验节点
 CM: 信道信息

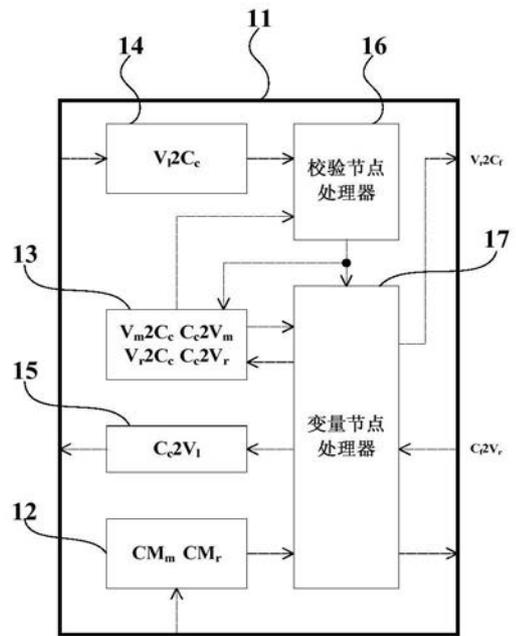
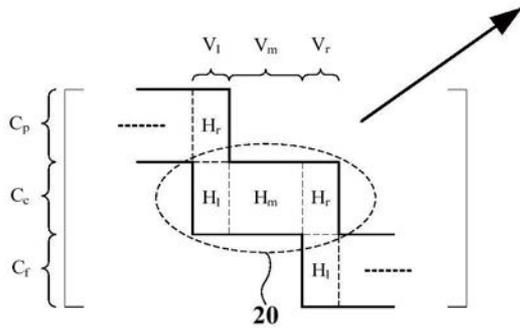


图8

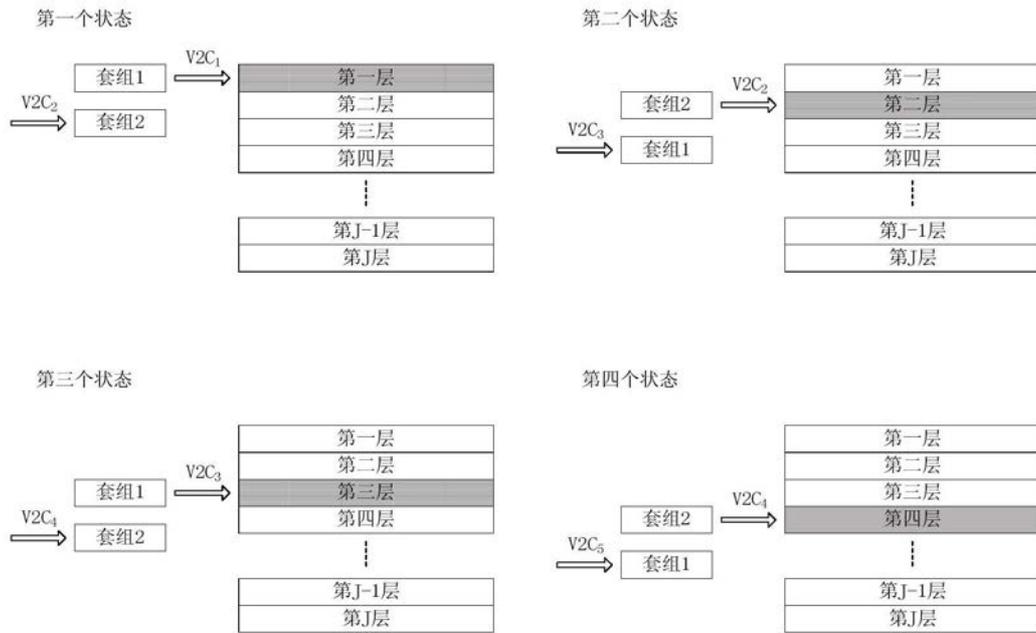


图9

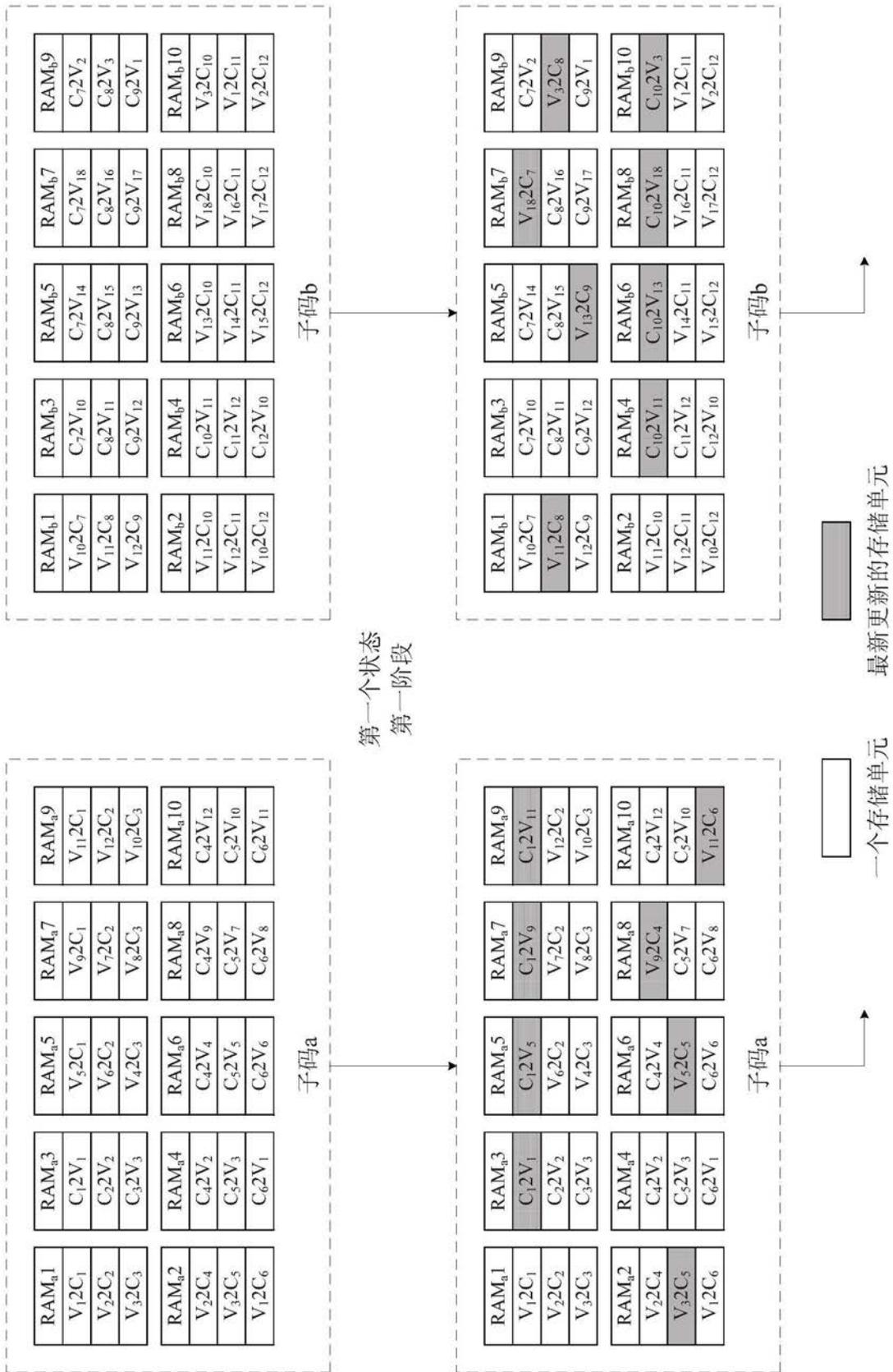


图11A

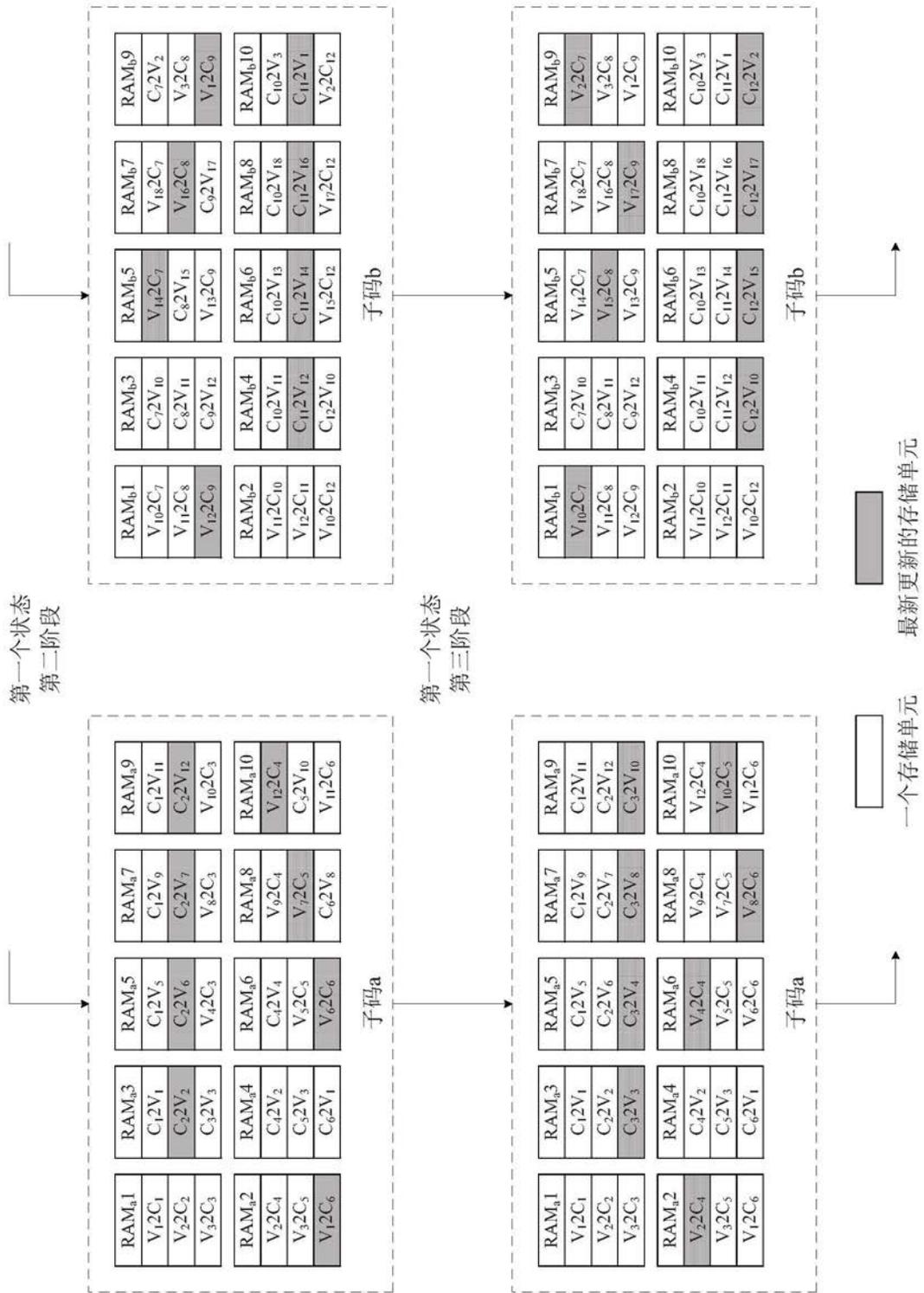


图11B

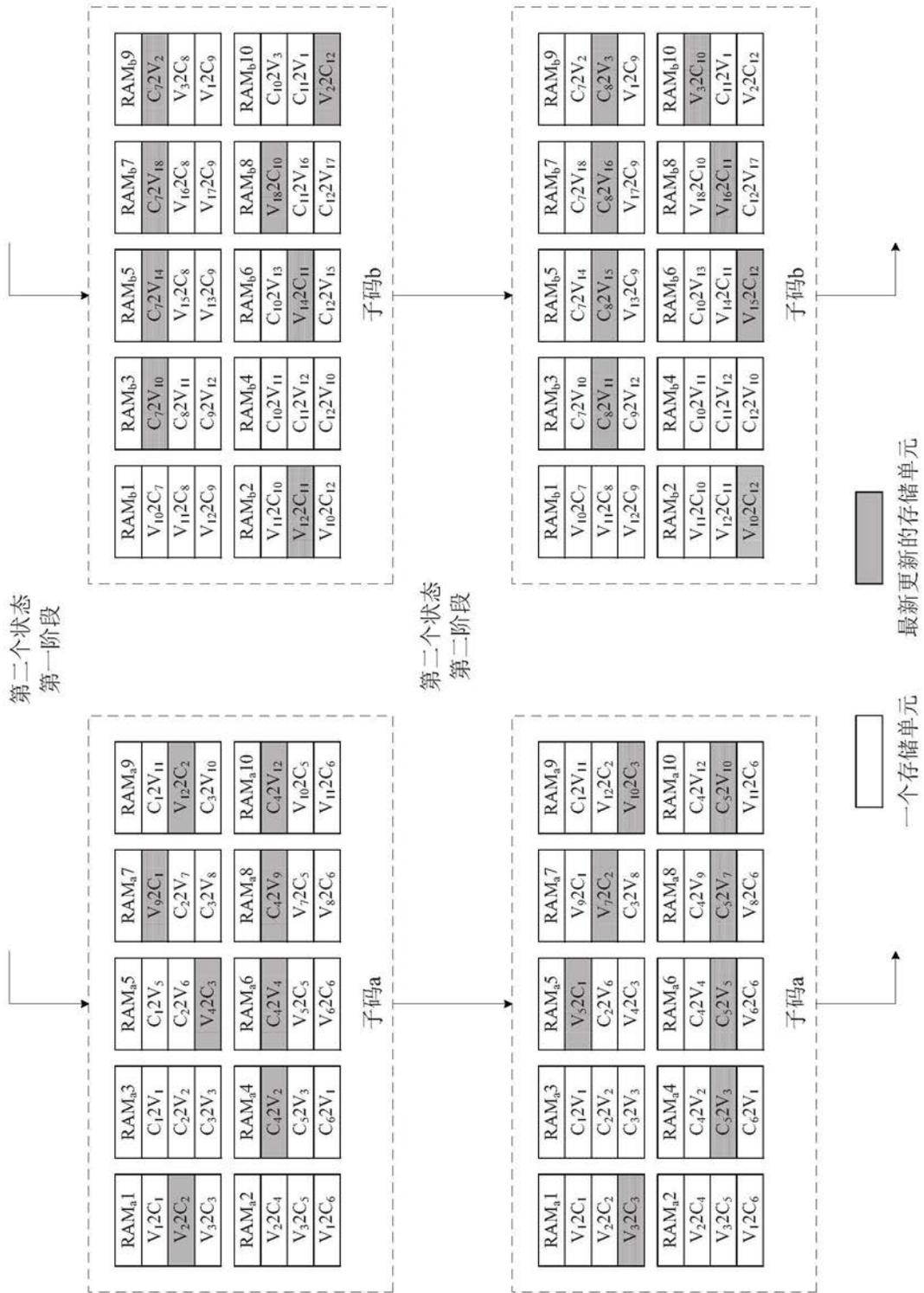


图11C

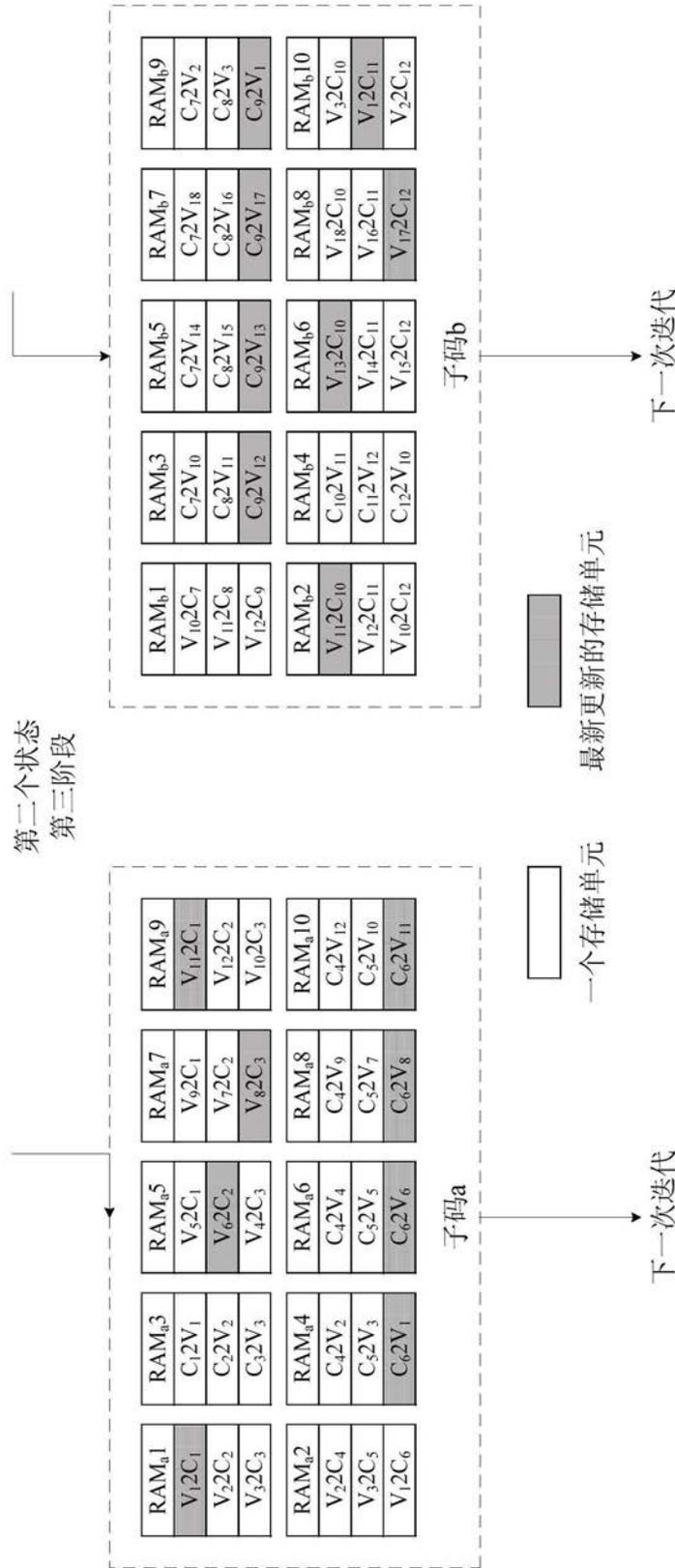


图11D

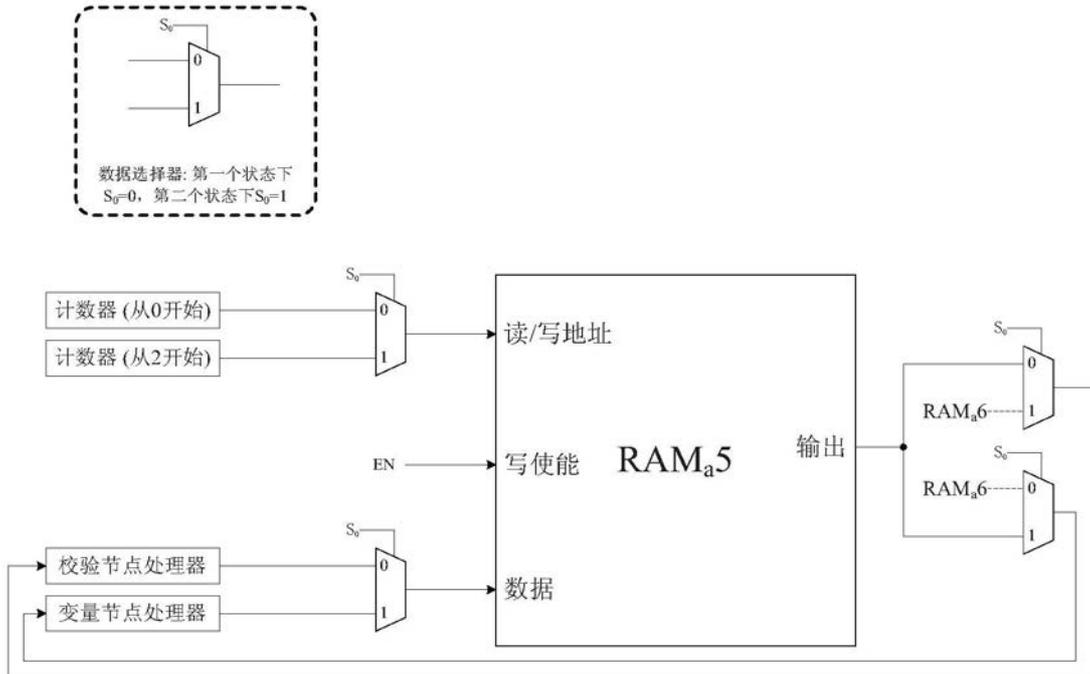


图12

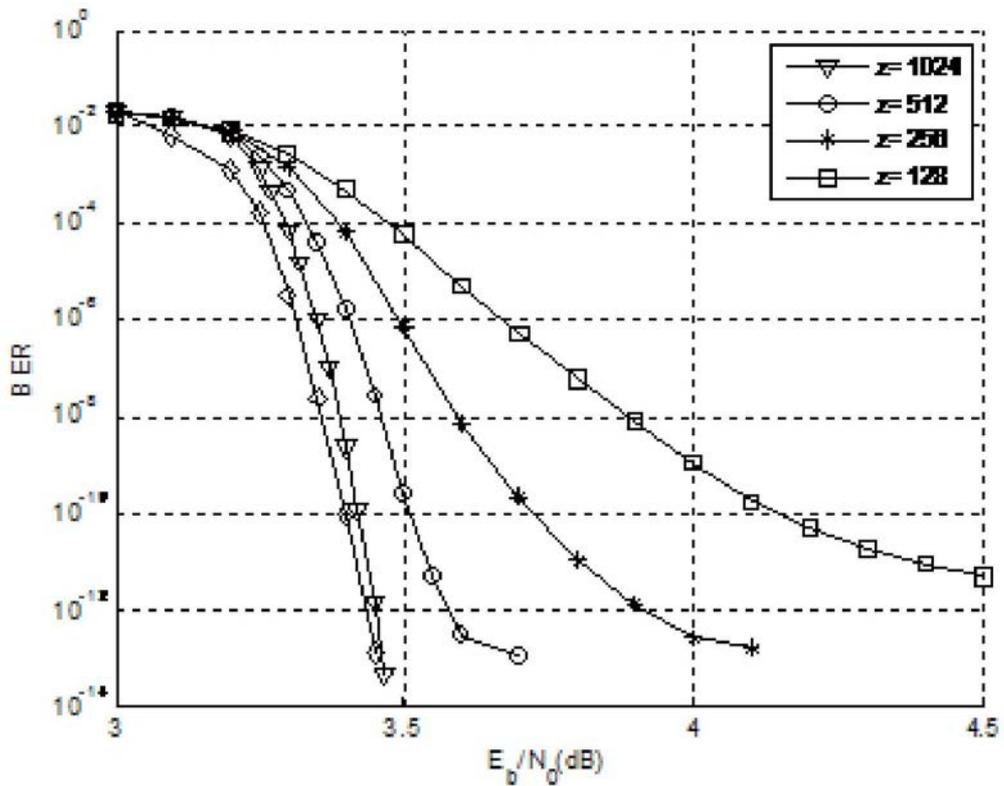


图13