



(12) 发明专利申请

(10) 申请公布号 CN 103069720 A

(43) 申请公布日 2013. 04. 24

(21) 申请号 201180038008. 1

代理人 宋巧苓

(22) 申请日 2011. 08. 03

(51) Int. Cl.

(30) 优先权数据

H03M 13/19 (2006. 01)

2010-177538 2010. 08. 06 JP

(85) PCT申请进入国家阶段日

2013. 02. 01

(86) PCT申请的申请数据

PCT/JP2011/004396 2011. 08. 03

(87) PCT申请的公布数据

W02012/017652 JA 2012. 02. 09

(71) 申请人 松下电器产业株式会社

地址 日本大阪府

(72) 发明人 四十九直也 冈村周太

(74) 专利代理机构 北京市柳沈律师事务所

11105

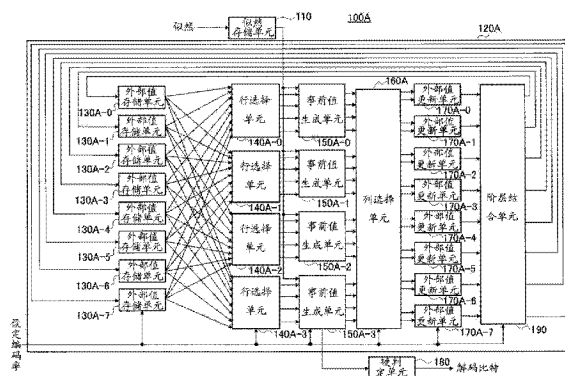
权利要求书2页 说明书18页 附图15页

(54) 发明名称

纠错解码装置及纠错解码方法

(57) 摘要

既能共用电路以抑制电路规模的增大,又能进行与多个编码率对应的LDPC (Low-DensityParity-Check,低密度奇偶校验)解码。当设定编码率为编码率比第1编码率大的第2编码率时,列处理行处理运算单元(120A)使用从与对应于第2编码率的第2校验矩阵相应的分散校验矩阵中,选择与构成第1部分矩阵的列数相同数的列并组合而成的分散部分矩阵。这里,分散校验矩阵是扩展第2校验矩阵的行数,并将第2校验矩阵中的第2校验矩阵的行权重大的行的元素分散配置到该行与扩展行所得的矩阵。此时,列处理行处理运算单元(120A)使用行权重为第1部分矩阵的行权重以下的分散部分矩阵。



1. 纠错解码装置,对通过从多个编码率中设定的设定编码率进行了低密度奇偶校验编码的编码比特进行解码,包括:

存储单元,存储接收所述编码比特而获得的似然;

运算单元,使用所述似然与由规定的校验矩阵生成的第1部分矩阵,反复进行列处理及行处理以计算软判定值;以及

判定单元,使用所述软判定值判定解码比特,其中,

所述第1部分矩阵是从所述规定的校验矩阵中针对每列选择所述各部分矩阵的各行权重为规定数以下的组合而得到的矩阵。

2. 如权利要求1所述的纠错解码装置,

所述运算单元使用从追加有用于分散所述校验矩阵的非零元素的行的第2校验矩阵中针对每列选择而生成的第2部分矩阵,反复进行列处理及行处理以计算软判定值。

3. 如权利要求2所述的纠错解码装置,还包括:

阶层结合器,基于通过最小和解码处理中的行处理获得的值的最小值即对数外部值,选择分散有所述校验矩阵的非零元素的行中的任一个。

4. 如权利要求2所述的纠错解码装置,

当所述规定的校验矩阵为多个时,所述第2校验矩阵的行数为行数最多的校验矩阵的行数以下。

5. 如权利要求2所述的纠错解码装置,

所述运算单元使用列数与所述第1部分矩阵的列数相同数的矩阵作为所述第2部分矩阵。

6. 如权利要求4所述的纠错解码装置,

所述规定的校验矩阵中包含列权重比所述第2校验矩阵的列权重大的校验矩阵。

7. 如权利要求2所述的纠错解码装置,

所述规定的校验矩阵包含与设定编码率相应的多个矩阵,

所述运算单元包括:

外部值存储单元,输出通过最小和解码处理中的行处理获得的值的最小值即对数外部值比;

行设定单元,根据所述设定编码率,设定与所述第1部分矩阵或所述第2部分矩阵对应的所述对数外部值比;

事前值生成单元,根据所述设定编码率及所设定的所述对数外部值比,生成对数事前值比并重新排列;

列设定单元,根据所述设定编码率,设定与所述第1部分矩阵或所述第2部分矩阵对应的所述对数事前值比;

外部值更新单元,根据所设定的所述对数事前值比,更新所述对数外部值比;以及

硬判定单元,基于所述对数事前值比判定解码比特。

8. 如权利要求7所述的纠错解码装置,

所述外部值更新单元的输入数为所述第1部分矩阵的行权重数以下。

9. 纠错解码方法,对通过从多个编码率中设定的设定编码率进行了低密度奇偶校验编码的编码比特进行解码,

该纠错解码方法是

存储接收所述编码比特而获得的似然，

使用所述似然与由规定的校验矩阵生成的部分矩阵，反复进行列处理及行处理以计算软判定值，

并使用所述软判定值判定解码比特的方法，

所述第 1 部分矩阵是从所述规定的校验矩阵中针对每列选择所述各部分矩阵的各行权重为规定数以下的组合而得到的矩阵。

纠错解码装置及纠错解码方法

技术领域

[0001] 本发明涉及一种使用低密度奇偶校验(LDPC:Low-Density Parity-Check)码的纠错解码装置及纠错解码方法。

背景技术

[0002] 近年来,作为发挥高纠错能力的纠错码,LDPC 码正备受瞩目。LDPC 码是以低密度的奇偶校验矩阵定义的纠错码。

[0003] LDPC 码的纠错能力高且容易安装,因此正被探讨作为 IEEE802.11n 的高速无线 LAN(Local Area Network,局域网)系统、数字广播系统或者大容量存储装置等中的纠错编码方式。

[0004] 作为以往的 LDPC 码的解码装置,例如有专利文献 1 所示的解码装置。专利文献 1 中记载的解码装置是与多个校验矩阵对应的解码装置,以单位区域中包含边缘的方式划分校验矩阵。这里,所谓边缘,是指校验矩阵的元素“1”。在二进制表达的 LDPC 码的情况下,校验矩阵的元素为“0”或“1”。所谓矩阵的元素,是指矩阵的成分。并且,专利文献 1 中记载的解码装置通过保存存在于所划分的群组内的边缘配置信息,从而进行存储器容量的削减。另外,专利文献 1 中公开了使用边缘配置信息简化存储器与运算器间的连接的方法。

[0005] 现有技术文献

[0006] 专利文献

[0007] 专利文献 1:特开 2007-215089 号公报

[0008] 非专利文献

[0009] 非专利文献 1:和田山正著“低密度パリティ検査符号とその復号法(低密度奇偶校验码及其解码法)”,トリケップス出版,2002 年 6 月 5 日(P92-P99)

发明内容

[0010] 发明要解决的问题

[0011] 但是,上述以往的结构中,解码处理中的运算器的安装数必须为行权重 $\times$ 同时处理行数。因此,在构成与多个校验矩阵对应的解码装置时,需要能够应对行权重最大的校验矩阵的数量的运算器,从而存在电路规模增大的问题。

[0012] 本发明鉴于此点而完成,其目的在于提供一种纠错解码装置及纠错解码方法,既能够共用电路以抑制电路规模的增大,又能够进行与多个编码率对应的 LDPC 解码。

[0013] 解决问题的方案

[0014] 本发明的纠错解码装置的一个形态是纠错解码装置,对于通过从多个编码率中设定的设定编码率进行了低密度奇偶校验编码的编码比特,使用与所述设定编码率对应的校验矩阵进行解码,该纠错解码装置包括:存储单元,存储接收所述编码比特而获得的似然;运算单元,使用所述似然及与对应于所述设定编码率的校验矩阵相应的部分矩阵,反复进行列处理及行处理以计算软判定值;以及判定单元,使用所述软判定值判定解码比特,其

中,所述运算单元在所述设定编码率为第 1 编码率时,使用根据所述列处理中的解码对象的列数而从与所述第 1 编码率对应的第 1 校验矩阵中选择任意列组合而成的第 1 部分矩阵来作为所述部分矩阵,在所述设定编码率为编码率比所述第 1 编码率大的第 2 编码率时,使用根据构成所述第 1 部分矩阵的列数而从与对应于所述第 2 编码率的第 2 校验矩阵相应的分散校验矩阵中选择任意列组合而成的分散部分矩阵来作为所述部分矩阵,所述分散校验矩阵是扩展所述第 2 校验矩阵的行数,并将所述第 2 校验矩阵中的所述第 2 校验矩阵的行权重大的行的元素分散配置到该行与扩展行所得的矩阵。

[0015] 本发明的纠错解码方法的一个形态是纠错解码方法,对于通过从多个编码率中设定的设定编码率进行了低密度奇偶校验编码的编码比特,使用与所述设定编码率对应的校验矩阵进行解码,该纠错解码方法是存储接收所述编码比特而获得的似然,使用所述似然及与对应于所述设定编码率的校验矩阵相应的部分矩阵,反复进行列处理及行处理以计算软判定值,并使用所述软判定值判定解码比特的方法,在所述设定编码率为第 1 编码率时,使用根据所述列处理中的解码对象的列数而从与所述第 1 编码率对应的第 1 校验矩阵中选择并组合而成的第 1 部分矩阵来作为所述部分矩阵,在所述设定编码率为编码率比所述第 1 编码率大的所述第 2 编码率时,使用根据构成所述第 1 部分矩阵的列数而从与对应于所述第 2 编码率的第 2 校验矩阵相应的分散校验矩阵中选择并组合而成的分散部分矩阵来作为所述部分矩阵,所述分散校验矩阵是扩展所述第 2 校验矩阵的行数,并将所述第 2 校验矩阵中的所述第 2 校验矩阵的行权重大的行的元素分散配置到该行与扩展行所得的矩阵。

[0016] 由此,既能够共用电路以抑制电路规模的增大,又能够进行与多个编码率对应的 LDPC 解码。

[0017] 发明的效果

[0018] 根据本发明,既能够共用电路以抑制电路规模的增大,又能够进行与多个编码率对应的 LDPC 解码。

附图说明

[0019] 图 1 是表示一例本发明的实施方式中的共用源的校验矩阵的图。

[0020] 图 2 是表示一例上述实施方式中的共用源的部分矩阵的图。

[0021] 图 3 是上述实施方式中的解码处理流程的图。

[0022] 图 4 是表示上述实施方式的共用源的解码器的主要部分结构的图。

[0023] 图 5 是表示一例上述实施方式中的共用化对象的校验矩阵的图。

[0024] 图 6 是表示一例上述实施方式中的共用对象的部分矩阵的图。

[0025] 图 7 是表示一例上述实施方式中的分散校验矩阵的图。

[0026] 图 8 是表示一例上述实施方式中的分散部分矩阵的图。

[0027] 图 9 是上述实施方式中的解码处理流程的图。

[0028] 图 10 是表示上述实施方式的共用解码器的主要部分结构的图。

[0029] 图 11 是表示上述实施方式的共用解码器的其他主要部分结构的图。

[0030] 图 12 是表示图 10 所示的解码器中的解码处理的时间图的图。

[0031] 图 13 是表示图 11 所示的解码器中的解码处理的时间图的图。

[0032] 图 14 是表示与多个编码率中的任意选择的编码率对应的 LDPC 解码装置的结构

图。

[0033] 图 15 是表示与第 1 编码率及第 2 编码率对应的 LDPC 解码装置的结构图。

[0034] 标号说明

[0035] 100、100A、100B 解码器

[0036] 110 似然存储单元

[0037] 120、120A、120B 列处理行处理运算单元

[0038] 130-0 ~ 130-7、130A-0 ~ 130A-7 外部值存储单元

[0039] 140-0 ~ 140-3、140A-0 ~ 140A-3 行选择单元

[0040] 150-0 ~ 150-3、150A-0 ~ 150A-3 事前值生成单元

[0041] 160、160A 列选择单元

[0042] 170-0 ~ 170-7、170A-0 ~ 170A-7、170B-0 ~ 170B-7 外部值更新单元

[0043] 180 硬判定单元

[0044] 190 阶层结合单元

具体实施方式

[0045] 以下,参照附图详细说明本发明的实施方式。

[0046] (实施方式)

[0047] 本实施方式对既能够共用电路以抑制电路规模的增大,又能够进行与多个编码率对应的 LDPC 解码的纠错解码装置(以下简称为“解码器”)及纠错解码方法(以下称为“LDPC 解码方法”或“解码算法”)进行说明。

[0048] 首先,对与多个编码率中的任意选择的编码率对应的 LDPC 解码方法进行说明。

[0049] 图 14 所示的解码器 100C 表示与多个编码率中的任意选择的编码率对应的 LDPC 解码装置的结构。向解码器 100C 输入由接收信号获得的被称为似然的信息。输入的似然被暂时存储在似然存储单元 110 中。列处理行处理运算单元 120 读出存储在似然存储单元 110 中的似然,进行解码处理。

[0050] 列处理行处理运算单元 120 输出在解码处理过程中获得的信息即外部值 200。列处理行处理运算单元 120 再次使用输出的外部值 200 作为输入信号。此外,图 14 中,从列处理行处理运算单元 120 暂时输出外部值 200,但未必需要输出,也能够列处理行处理运算单元 120 的内部进行处理。

[0051] 列处理行处理运算单元 120 使用外部值 200 及读出的似然反复进行解码处理(反复解码处理)。列处理行处理运算单元 120 在反复进行规定反复次数的解码处理后,向硬判定单元 180 输出事前值。硬判定单元 180 对输入的事前值进行硬判定,并输出解码比特。

[0052] 此外,关于似然、外部值 200、事前值,在非专利文献 1 中已详细记载。

[0053] 下面叙述列处理行处理运算单元 120 中的 LDPC 解码方法的细节。

[0054] 图 1 是表示与多个编码率中的任意选择的编码率对应的 LDPC 码的校验矩阵的图。

[0055] 在图 1 中,由四方框围成的部分相当于子矩阵,第 1 校验矩阵是结合多个子矩阵而构成。该子矩阵是对单位矩阵进行循环移位所得的矩阵,在图 1 中,由四方框围成的数值表示单位矩阵的循环移位量。

[0056] 例如,0 行 0 列的循环移位量“0”表示单位矩阵其自身,1 行 2 列的循环移位量“9”

表示对单位矩阵进行 9 循环移位所得的矩阵。另一方面，“-”表示零矩阵，并非单位矩阵的循环移位。

[0057] 单位矩阵的大小为 4×4 ，循环移位量为 2 的子矩阵如式(1)表达。由式(1)可知，循环移位量为 2 的子矩阵是将单位矩阵的元素“1”向右移位 2 位所得的矩阵。

$$[0058] \quad I_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \dots\dots (1)$$

[0059] 以下，作为一例，以图 1 的校验矩阵中的子矩阵大小为 64×64 的情况为例进行说明。此时，例如，0 行 2 列的子矩阵是将 64×64 的单位矩阵向右进行 8 循环移位的矩阵，2 行 3 列的子矩阵是将 64×64 的单位矩阵向右进行 12 循环移位的矩阵。

[0060] 图 1 的校验矩阵中，列数为 16 列，因此码长为 $16 \times 64 = 1024$ ，行数为 8 行，因此奇偶长为 $8 \times 64 = 512$ 。因而，信息比特长为 512 比特，图 1 的校验矩阵为编码率 $1/2$ ($R = 1/2$) 的校验矩阵。此外，本实施方式中的编码率及校验矩阵并无特别限定，使用图 1 所示的编码率 $1/2$ 的校验矩阵为例进行说明。

[0061] 本实施方式中，并非是对校验矩阵的所有列数同时进行与校验矩阵对应的 LDPC 解码，而是分割成每若干列数来进行。具体而言，本实施方式中，将校验矩阵分割成多个部分矩阵，对每个部分矩阵进行解码处理。

[0062] 以下，作为一例，考虑对于与由 16 列构成的图 1 的校验矩阵对应的 LDPC 解码，每 4 列数进行解码处理的情况。

[0063] 图 2 是表示对每 4 列数进行解码处理时所用的部分矩阵的图。图 2 是对于如图 1 所示那样分配了编号的列，如 $\{0, 1, 2, 3\}$ 、 $\{4, 5, 6, 7\}$ 、 $\{8, 9, 10, 12\}$ 、 $\{11, 13, 14, 15\}$ 地选择组合而形成部分矩阵的例子。图 1 所示的例子中，由 16 列构成的校验矩阵被分割成由 4 列构成的部分矩阵。此外，列编号 12 被选择为部分矩阵 # 1-2，列编号 11 被选择为部分矩阵 # 1-3。

[0064] 用于由校验矩阵形成部分矩阵的列选择的规范如下。

[0065] 本实施方式中，以部分矩阵的行权重数为最小的方式，从校验矩阵中选择列来生成部分矩阵。在图 2 中，与各部分矩阵的各行对应的行权重全部为 2 以下。这里，所谓行权重，表示矩阵的各行中的元素“1”的数量。即，当行权重为 2 时，表示当沿行方向对矩阵的元素“1”的数量进行计数时，存在两个元素“1”的事实。在图 2 中，表示记载了循环移位量的子矩阵的个数在所有行中为两个以下。

[0066] 另外，当对每 2 列数进行解码处理时，对于图 1 的校验矩阵，例如着眼于列编号 9、12、13，如 $\{0, 1\}$ 、 $\{2, 3\}$ 、 $\{4, 5\}$ 、 $\{6, 7\}$ 、 $\{8, 10\}$ 、 $\{9, 12\}$ 、 $\{11, 14\}$ 、 $\{13, 15\}$ 地每两列地选择列来形成部分矩阵。由此，能够将行权重设为 1。此外，对于构成部分矩阵的列数，即，统一进行解码处理的列数的设定，将在后面叙述。

[0067] 这样，本实施方式中，根据进行解码处理的列数，即，构成部分矩阵的列数，以部分矩阵的行权重数为最小的方式选择列，从而由校验矩阵生成部分矩阵。

[0068] 本实施方式中，按照上述列选择来形成部分矩阵，并且进行以下的解码处理。以下，作为本解码算法，例如对使用对数区域最小和(min-sum)解码的情况进行说明。通过参

照非专利文献 1, 能够获得对数区域最小和解码的细节。本实施方式中, 以如下方式执行对数区域最小和解码。

[0069] 表 1 表示本解码算法中所用的变量的定义。

[0070]

变量名	标注	详细
更新处理实施次数	i	进行 LDPC 解码中的更新处理的次数
行编号	m (图 1 的情况下为 0、1、.....、7)	如图 1 那样特定子矩阵的行编号
列编号	n (图 1 的情况为 0、1、.....、15)	如图 1 那样特定子矩阵的列编号
子矩阵内索引	z (子矩阵为 64×64 的情况下为 0、1、.....、63)	特定子矩阵内的行或列的编号

[0071]

对数似然比	$\lambda(n, z)$	与 LDPC 码比特（本实施方式中为 1024（=16x64）比特）对应的对数似然比。与第 n 列的子矩阵的第 z 列对应。
输出软判定值	$\Lambda_s(n, z)$	将与 LDPC 码比特对应的输出软判定值标记为第 n 列子矩阵的第 z 列。
输出硬判定值	$\Lambda_h(n, z)$	对输出软判定值进行硬判定的值
对数外部值比 （与第 i 次更新处理中的第 m 行子矩阵的 z 行对应的外部值）	$\alpha_{\min}(i, m, z)$	外部值更新时，对数外部值比的绝对值的最小值
	$\alpha_{2nd}(i, m, z)$	外部值更新时，第 2 个小的对数外部值比的绝对值
	$\alpha_{\text{ind}}(i, m, z)$	外部值更新时，赋予对数外部值比的绝对值的最小值的子矩阵的列编号
	$\alpha_{\text{sig}}(i, m, z)$	对数外部值比的符号
对数事前值比	$\beta(m, n, z)$	与第 m 行第 n 列子矩阵的第 z 列对应的对数事前值比
	$\beta_{\text{sig}}(m, n, z)$	与第 m 行第 n 列子矩阵的第 z 列对应的对数事前值比的符号

[0072] 如图 2 所示，本解码算法以从校验矩阵选择列而形成的部分矩阵的列数为单位进行行列处理。而且，本解码算法使用作为列处理的结果而获得的对数事前值比，逐次进行行处

理。

[0073] 图 2 的例子中,本解码算法进行 4 次每 4 列(每部分矩阵的列数)的列处理。具体而言,依图 2 所示的部分矩阵 # 1-0、部分矩阵 # 1-1、部分矩阵 # 1-2、部分矩阵 # 1-3 的顺序进行列处理。这样,本解码算法在实施了 4 次每 4 列(每部分矩阵的列数)的列处理的时刻,等同于实施 1 次使用图 1 所示的校验矩阵的列处理的更新。

[0074] 图 3 是表示本实施方式的 LDPC 解码处理的流程的图。以下,使用图 3 所示的流程说明本解码算法的细节。

[0075] [1] 开始

[0076] 获得 LDPC 码的所有比特量(本实施方式中为 1024 比特)的对数似然比后,本解码算法开始 LDPC 解码。

[0077] [2] 初始设定(ST101)

[0078] 本解码算法将更新处理次数设定为第 0 次。另外,本解码算法将第 0 次更新处理中的所有相对于 m, z 的对数外部值比设定为如下。这里, m 是指如图 1 那样以子矩阵(子块)表示校验矩阵时的行编号。另外, z 是将通过 m 选择的子矩阵(子块)转换成单位矩阵的循环移位时的行的索引。

[0079] 此外,对数外部值(外部值)表示在反复解码中解码对象比特的解码可靠度的增量。对于最小和解码而言,对数外部值(外部值)是指通过使用与解码对象比特以外的比特对应的事前值进行的行处理而获得的可靠度的增量。

[0080] $\alpha_{\min}(0, m, z) = 0 \cdots \cdots (2-1)$

[0081] $\alpha_{2nd}(0, m, z) = 0 \cdots \cdots (2-2)$

[0082] $\alpha_{ind}(0, m, z) = 0 \cdots \cdots (2-3)$

[0083] $\alpha_{sig}(0, m, z) = 1 \cdots \cdots (2-4)$

[0084] 另外,本解码算法对于所有 m, n, z ,将对数事前值比的符号设定为如下。这里, n 是指如图 1 那样以子矩阵(子块)表示校验矩阵时的列编号。

[0085] $\beta_{sig}(m, n, z) = 1 \cdots \cdots (3)$

[0086] [3] $i = 1$ (ST102)

[0087] 本解码算法中,将更新处理实施次数计数用变量(以下称为“更新次数”) i 设置为 1。

[0088] [4] 外部值初始化(ST103)

[0089] 本解码算法对于所有 m, z ,将第 i 次更新处理中的对数外部值比设定为如下。

[0090] $\alpha_{\min}(i, m, z) = qt_{\max} - 1 \cdots \cdots (4-1)$

[0091] $\alpha_{2nd}(i, m, z) = qt_{\max} \cdots \cdots (4-2)$

[0092] $\alpha_{ind}(i, m, z) = 0 \cdots \cdots (4-3)$

[0093] $\alpha_{sig}(i, m, z) = 1 \cdots \cdots (4-4)$

[0094] 其中, $qt_{\max}$ 表示处理系中的最大值。

[0095] [5] $j = 0$ (ST104)

[0096] 本解码算法中,将表示解码顺序的变量 j 设置为 0。此外,解码顺序 j 如图 2 所示,是对部分矩阵(实施解码处理的列的集合)附加的编号。

[0097] [6] 计算事前值(列处理) (ST105)

[0098] 首先,本解码算法中,对于所有 m 、 n 、 z_0 ,根据第 $(i-1)$ 次更新处理中求出的对数外部值比,以如下方式求出用于与部分矩阵对应的列处理的输入外部值 $e_{val}(m, n, z_1)$ 。

$$[0099] \quad e_{sig}(m, n, z_1) = \alpha_{sig}(i-1, m, z_0) \times \beta_{sig}(m, n, z_1) \cdots \cdots (5)$$

[0100] 这里, Z 为子矩阵大小, $S(m, n)$ 为校验矩阵相对于第 m 行第 n 列子矩阵的循环移位量。

[0101] 另外, $z_0=0, \cdots, Z-1$, 且 $z_1 = (S(m, n) + z_0) \bmod Z$ 。其中, $X \bmod Y$ 表示 Y 相对于 X 的余数。

$$[0102] \quad e_{abs}(m, n, z_1) = \alpha_{min}(i-1, m, z_0) (\alpha_{ind}(i-1, m, z_0) \neq n \text{ 时}) \cdots \cdots (6-1)$$

$$[0103] \quad \alpha_{2nd}(i-1, m, z_0) (\alpha_{ind}(i-1, m, z_0) = n \text{ 时}) \cdots \cdots (6-2)$$

$$[0104] \quad e_{val}(m, n, z_1) = e_{sig}(m, n, z_1) \times e_{abs}(m, n, z_1) \cdots \cdots (7)$$

[0105] 但是,本解码算法中,当 $S(m, n) = "-"$ 时,即, $S(m, n)$ 所示的子矩阵为零矩阵时,不计算输入外部值。其中, $S(m, n)$ 表示校验矩阵相对于第 m 行第 n 列子矩阵的循环移位量。此外,子矩阵内索引相对于对数外部值比的索引 z_0 与相对于输入外部值的索引 z_1 不同,是因为处理顺序因子矩阵的循环移位而调换。

[0106] 使用以上述方式获得的输入外部值与对数似然比,本解码算法求出对数事前值比。针对所有 m 、 n 、 z ,如式(8)那样求出对数事前值比。

$$[0107] \quad \beta(m, n, z) = \lambda(n, z) + \sum_{m'=0 \setminus m}^{M-1} e_{val}(m', n, z) \cdots \cdots (8)$$

[0108] 其中, M 表示校验矩阵的子矩阵行数。

[0109] 式(8)中,对除了子矩阵的行编号 m 以外的子矩阵进行 Σ 。此外,在式(8)中, $0 \setminus m$ 表示“除了行编号 m 以外”。而且,为了生成下次更新处理中的输入外部值 $e_{val}(m, n, z_1)$,如式(9)那样更新对数事前值比的符号。

$$[0110] \quad \beta_{sig}(m, n, z) = \text{SIGN}(\beta(m, n, z)) \cdots \cdots (9)$$

[0111] 其中, $\text{SIGN}(\cdot)$ 为返回符号的函数。

[0112] [7] 计算外部值(行处理)(ST106)

[0113] 本解码算法中,对由如图2那样选择的子矩阵的集合构成的部分矩阵进行逐次行处理。最小和解码中的行处理是探索最小值的运算。因此,本解码算法中,将[6]中求出的对数事前值比与至此为止求出的对数外部值比进行比较,探索绝对值的最小值,并将探索出的最小值更新为对数外部值比。

[0114] 另外,本解码算法在最小值探索中,还求出赋予绝对值的最小值的列编号及绝对值为第2个小的值(以下称为“第二值”),以便用于前述的列处理。而且,本解码算法中,还更新对数外部值比的符号。此外,这些行处理通过以下的数式表达。

$$[0115] \quad \beta_{min}(m, z_1) = \min_n (\beta(m, n, z_1)) \cdots \cdots (10-1)$$

$$\beta_{ind}(m, z_1) = \min_{ind} (\beta(m, n, z_1)) \cdots \cdots (10-2)$$

[0116] 其中, $\min(\cdot)$ 表示返回沿行方向观察与解码对象的列对应的对数事前值比时的绝对值的最小值的函数。

[0117] 另外, $\min_{ind}(\cdot)$ 是返回赋予上述绝对值的最小值的子矩阵的列编号的函数。

$$[0118] \quad z_1 = (S(m, n) + z_0) \bmod Z \cdots \cdots (11-1)$$

[0119] $\alpha_{\min}(i, m, z_0) = \min(\alpha_{\min}(i, m, z_0), \beta_{\min}(m, z_1)) \dots\dots (11-2)$

[0120] $\alpha_{2nd}(i, m, z_0) = \min2(\alpha_{\min}(i, m, z_0), \alpha_{2nd}(i, m, z_0), \beta_{\min}(m, z_1)) \dots\dots (11-3)$

[0121] $\alpha_{ind}(i, m, z_0) = \minind(\alpha_{ind}(i, m, z_0), \beta_{ind}(m, z_1)) \dots\dots (11-4)$

[0122] 其中, $\min(\cdot)$ 表示返回最小值的函数, $\min2(\cdot)$ 是返回第二值的函数。

[0123] 另外, $\minind(\cdot)$ 是在通过 $\min$ 函数选择 $\beta_{\min}(m, z_1)$ 作为最小值的情况下返回 $\beta_{ind}(m, z_1)$, 而在除此以外的情况下返回 $\alpha_{ind}(i, m, z_0)$ 的函数。

[0124] [8] $j += 1$ (ST107)

[0125] 本解码算法将解码顺序 j 增 1。

[0126] [9] $j \leq J$ (ST108)

[0127] 本解码算法中, 若 j 为最后的解码顺序 J 以下, 则返回 [6] 计算事前值而执行列处理, 若 j 超过 J , 则退出循环。

[0128] [10] $i += 1$ (ST109)

[0129] 本解码算法将更新次数 i 增 1。

[0130] [11] $i < itmax$ (ST110)

[0131] 本解码算法中, 若更新次数 i 小于最大更新次数 $itmax$, 则返回 [4] 外部值初始化而再次开始解码处理。另一方面, 若更新次数 i 为最大更新次数 $itmax$ 以上, 则退出循环。

[0132] [12] 计算输出硬判定值 (ST111)

[0133] 本解码算法中, 按照下式对所有 n, z 求出输出软判定值。

$$[0134] \quad \Lambda_s(n, z) = \lambda(n, z) + \sum_{m'=0}^{M-1} e_{val}(m', n, z) \quad \dots\dots (12)$$

[0135] 求出对数事前值比时, 本解码算法如式 (8) 所示, 未累加与解码对象的行编号 m 的子矩阵对应的 e_{val} 。但是, 在输出软判定值的计算过程中, 本解码算法也将解码对象的行编号 m 的子矩阵考虑在内。

[0136] 本解码算法使用所获得的输出软判定值的符号来求出输出硬判定值。此外, 输出软判定值的符号及输出硬判定值取决于对数似然比的求法。当如非专利文献 1 那样求出对数似然比时, 本解码算法中, 若输出软判定值的符号为正, 则输出“0”作为解码结果, 若为负, 则输出“1”。并且, 本解码算法输出与信息比特对应的解码结果作为解码比特, 并结束 LDPC 解码。

[0137] 图 4 是表示本实施方式的解码器的主要部分结构的方块图。图 4 的解码器 100 是使用 LDPC 校验矩阵进行纠错解码的纠错解码装置, 基于图 3 所示的流程进行 LDPC 解码。此外, 图 4 中, 作为一例, 表示与图 1 所示的校验矩阵对应的解码器 100 的结构。另外, 图 4 中, 为了简化记载, 将子矩阵作为处理单位来记载解码器 100 的结构。

[0138] 本实施方式的解码器 100 具有似然存储单元 110、列处理行处理运算单元 120 及硬判定单元 180。

[0139] 列处理行处理运算单元 120 具有外部值存储单元 130-0 ~ 130-7、行选择单元 140-0 ~ 140-3、事前值生成单元 150-0 ~ 150-3、列选择单元 160 及外部值更新单元 170-0 ~ 170-7。

[0140] 图 1 的校验矩阵的部分矩阵 (参照图 2) 是以子矩阵为单位的 8 行 $\times$ 4 列的矩阵。即, 本解码算法对于子矩阵的 8 行统一进行 LDPC 解码中的行处理。因此, 图 4 的解码器 100

采用具有 8 个外部值存储单元及外部值更新单元的结构。另外,本解码算法对于子矩阵的 4 列统一进行 LDPC 解码中的列处理。因此,图 4 的解码器 100 采用具有 4 个矩阵选择单元及事前值生成单元的结构。

[0141] 此外,构成部分矩阵的列数并不限于 4,解码器 100 只要具有与构成部分矩阵的列数相应个数的矩阵选择单元及事前值生成单元即可。构成部分矩阵的列数越大,所需要的矩阵选择单元及事前值生成单元就越多,因此要兼顾电路规模来设定构成部分矩阵的列数。

[0142] 似然存储单元 110 存储由接收信号求出的与 LDPC 码比特对应的对数似然比。一旦在似然存储单元 110 中存储 LDPC 码的所有比特量的对数似然比,则开始 LDPC 解码。

[0143] 外部值存储单元 130-k ($k = 0 \sim 7$) 存储从外部值更新单元 170-k 输出的对数外部值比。此外,在 LDPC 解码开始时,外部值存储单元 130-k ($k = 0 \sim 7$) 如上述流程图“[2] 初始设定”中所述地,将对数外部值比初始化。

[0144] 外部值存储单元 130-k 将存储的对数外部值比输出至行选择单元 140-0 ~ 140-3。

[0145] 行选择单元 140-r ($r = 0 \sim 3$) 选择与图 2 所示的解码顺序 j 的部分矩阵 # 1-j (4 列解码单位) 的列编号 r 的列中的非零元素的行(记载了循环移位量的行)对应的对数外部值比并输出。例如,行选择单元 140-0 选择与部分矩阵 # 1-0 的列编号 0 的非零元素(第 0、1、4、5 行)对应的对数外部值比,并输出到事前值生成单元 150-0。同样地,行选择单元 140-1 选择与部分矩阵 # 1-0 的列编号 1 的非零元素(第 2、3、6、7 行)对应的对数外部值比,并输出到事前值生成单元 150-1。行选择单元 140-2 选择与部分矩阵 # 1-0 的列编号 2 的非零元素(第 0、1、4、5 行)对应的对数外部值比,并输出到事前值生成单元 150-2。行选择单元 140-3 选择与部分矩阵 # 1-0 的列编号 3 的非零元素(第 2、3、6、7 行)对应的对数外部值比,并输出到事前值生成单元 150-3。

[0146] 事前值生成单元 150-r ($r = 0 \sim 3$) 使用从行选择单元 140-r 输入的对数外部值比与从似然存储单元 110 输入的对数似然比,如上述流程图“[6] 计算事前值(列处理)”所示那样求出对数事前值比。

[0147] 这里,事前值生成单元 150-r ($r = 0 \sim 3$) 求出与部分矩阵 # 1-j (解码对象的列)的列编号 r 的列中的非零元素的行对应的对数事前值比。例如,事前值生成单元 150-0 求出与部分矩阵 # 1-0 的列编号 0 的非零元素的行(第 0、1、4、5 行)对应的对数事前值比,并将求出的对数事前值比输出到列选择单元 160。同样地,事前值生成单元 150-1、150-2、150-3 分别求出与部分矩阵 # 1-j 的列编号 1、2、3 的非零元素对应的对数事前值比,并将求出的对数事前值比输出到列选择单元 160。

[0148] 此外,如数式(8)所示,通过加法求出对数事前值比。因此,事前值生成单元 150-r ($r = 0 \sim 3$) 在内部具有加法器。

[0149] 这里,事前值生成单元 150-3 在进行规定次数(itmax)的解码处理后,将对数事前值比输出至硬判定单元 180。

[0150] 列选择单元 160 选择与图 2 所示的解码顺序 j 的部分矩阵 # 1-j 的各行中的非零元素对应的对数事前值比。例如,列选择单元 160 考虑进行与解码顺序 0 的部分矩阵 # 1-0 相关的解码处理的情况。此时,关于部分矩阵 # 1-0 的第 0 行,列选择单元 160 选择与部分矩阵 # 1-0 的列编号 0、2 对应的对数事前值比。关于部分矩阵 # 1-0 的第 1 行,列选择单元

160 选择与部分矩阵 # 1-0 的列编号 0、2 对应的对数事前值比。另外,关于部分矩阵 # 1-0 的第 2 行,列选择单元 160 选择与部分矩阵 # 1-0 的列编号 1、3 对应的对数事前值比。另外,关于部分矩阵 # 1-0 的第 3 行,列选择单元 160 选择与部分矩阵 # 1-0 的列编号 1、3 对应的对数事前值比。

[0151] 这样,关于解码对象 j 的部分矩阵 # 1- j 的各行,列选择单元 160 选择与非零元素的列对应的对数事前值比,并将选择的对数事前值比输出到外部值更新单元 170- k ($k = 0 \sim 7$)。

[0152] 外部值更新单元 170- k ($k = 0 \sim 7$) 中,作为上述流程图“[7] 计算外部值(行处理)”,使用对数事前值比来更新与第 k 行对应的对数外部值比。例如,外部值更新单元 170-0 更新与部分矩阵 # 1- j 的第 0 行对应的对数外部值比。同样地,外部值更新单元 170- k ($k = 1 \sim 7$) 更新与部分矩阵 # 1- j 的第 k 行对应的对数外部值比。

[0153] 此外,如数式(10-1)、(10-2)及式(11-1)~(11-4)所示,通过比较求出对数外部值比。因此,外部值更新单元 170- k ($k = 0 \sim 7$) 在内部具有比较器。此外,比较器的数量必须为部分矩阵的行权重数个。

[0154] 硬判定单元 180 在进行规定次数(itmax)的解码处理后,对从事前值生成单元 150-3 输出的对数事前值比进行硬判定,获取硬判定值作为解码比特,并输出所获取的解码比特。

[0155] 此外,图 4 所示的解码器 100 示出了硬判定单元 180 对从事前值生成单元 150-3 输出的对数事前值比进行硬判定的结构,但并不限于此。即,也可采用硬判定单元 180 对从事前值生成单元 150-0、150-1、150-2 中的任意的从事前值生成单元输出的对数事前值比进行硬判定的结构。

[0156] 说明以上述方式构成的解码器 100 的动作。

[0157] 由接收信号求出与 LDPC 码比特对应的对数似然比,并存储到似然存储单元 110 中。一旦存储 LDPC 码的所有比特量的对数似然比,则开始 LDPC 解码。在 LDPC 解码开始时,如上述流程图“[2] 初始设定”那样,将存储在外部值存储单元 130- k ($k = 0 \sim 7$) 中的对数外部值比及存储在事前值生成单元 150- r ($r = 0 \sim 3$) 中的对数事前值比的符号初始化。

[0158] 在外部值存储单元 130- k ($k = 0 \sim 7$) 中初始化的对数外部值比被输入行选择单元 140- r ($r = 0 \sim 3$)。行选择单元 140- r ($r = 0 \sim 3$) 选择与图 2 所示的部分矩阵(4 列解码单位中)的第 r 列中的非零元素的行(记载了循环移位量的行)对应的对数外部值比并输出。

[0159] 所选择的对数外部值比被输入事前值生成单元 150- r ($r = 0 \sim 3$)。事前值生成单元 150- r ($r = 0 \sim 3$) 使用所输入的对数外部值比与从似然存储单元 110 输入的对数似然比,如上述流程图“[6] 计算事前值(列处理)”所示那样求出对数事前值比。求出的对数事前值比对应于解码对象的部分矩阵的列中的非零元素的行。

[0160] 从事前值生成单元 150- r ($r = 0 \sim 3$) 输出的对数事前值比被输入列选择单元 160。关于图 2 所示的由解码对象的 4 列构成的部分矩阵,列选择单元 160 选择与各行的非零元素对应的对数事前值比并输出。例如,当进行与解码顺序 0 的部分矩阵 # 1-1 相关的解码处理时,列选择单元 160 关于第 0 行,选择与列编号 0、2 对应的对数事前值比并输出。同

样地,列选择单元 160 关于第 1 行,选择与列编号 0、2 对应的对数事前值比并输出。另外,列选择单元 160 关于第 2 行,选择与列编号 1、3 对应的对数事前值比并输出。这样,列选择单元 160 具备关于解码对象的部分矩阵的各行,选择与非零元素的列对应的对数事前值比并输出的功能。

[0161] 在列选择单元 160 中选择的与各行的非零元素对应的对数事前值比被输入外部值更新单元 170-k ($k = 0 \sim 7$)。外部值更新单元 170-k ($k = 0 \sim 7$) 使用所输入的对数事前值比,如上述流程图“[7] 计算外部值(行处理)”所示那样更新与各行对应的对数外部值比。这样,外部值更新单元 170-k ($k = 0 \sim 7$) 具有更新与解码对象的部分矩阵的各行对应的对数外部值比的功能。

[0162] 以上的处理为图 2 中的解码顺序 j 中的解码处理,即,与部分矩阵 # 1-j 相关的解码处理。当与解码顺序 j 相关的解码处理结束时,开始与对应于下个解码顺序(j + 1)的部分矩阵相关的解码处理(流程图“[8]、[9]”)。

[0163] 这样,解码器 100 进行与解码顺序 j 对应的解码处理,并更新对数外部值比。当最后的解码顺序 J (图 2 中为解码顺序 J = 3) 完成时,外部值更新单元 170-k ($k = 0 \sim 7$) 输出更新后的对数外部值比。所输出的对数外部值比被输入外部值存储单元 130-k ($k = 0 \sim 7$)。外部值存储单元 130-k ($k = 0 \sim 7$) 存储在最后的解码顺序 J 中更新的对数外部值比。解码器 100 使用上述更新的对数外部值比再次进行同样的解码处理(流程图“[10]、[11]”)。

[0164] 并且,解码器 100 反复进行规定次数(itmax 次)的上述解码处理。在规定次数的解码处理后,如上述流程图“[12] 计算输出硬判定值”所示,事前值生成单元 150-3 求出输出软判定值并输出所求出的输出软判定值。

[0165] 硬判定单元 180 由输入的软判定值求出硬判定值,并作为解码比特而输出。

[0166] 这样,本实施方式中,解码器 100 对每个部分矩阵进行解码。此时,向外部值更新单元 170-k ($k = 0 \sim 7$) 输入的对数事前值比的数量与部分矩阵的行权重相同数。因此,通过以行权重为最小的方式分割校验矩阵以形成部分矩阵,从而能够使输入到外部值更新单元 170-k ($k = 0 \sim 7$) 的对数事前值比的数量为最小。由此,流程图“[7] 计算外部值(行处理)”中的用于最小值探索、第二值探索的输入数为最小,其结果,能够使构成外部值更新单元 170-k ($k = 0 \sim 7$) 的比较器的数量最小化。

[0167] 以上,对与多个编码率中的任意选择的编码率(以下称为“第 1 编码率”)对应的 LDPC 解码算法及解码器的结构进行了说明。接下来,对与编码率比第 1 编码率大的第 2 编码率对应的 LDPC 解码算法及解码器进行说明。

[0168] 图 15 所示的解码器 100A 是表示与第 1 编码率及第 2 编码率对应的 LDPC 解码装置的结构图。为了构成与第 1 编码率及第 2 编码率对应的解码器 100A,如图 15 所示需要设定编码率的输入与阶层结合单元 190。根据设定编码率,控制列处理行处理运算单元 120 及阶层结合单元 190 的处理内容。以下记载详细的解码处理。

[0169] 图 5 是表示第 2 编码率的校验矩阵(以下称为“第 2 校验矩阵”)的图。此外,图 5 的第 2 校验矩阵是编码率 5/8 ($R = 5/8$) 的校验矩阵。编码率是由(列数 - 行数)/(列数)决定的值,第 2 校验矩阵由 6 行 16 列的子矩阵集合构成,因此由第 2 校验矩阵定义的 LDPC 码的编码率为 $(16-6)/16 = 5/8$ 。这里,对于第 2 编码率的校验矩阵,使用列权重为第 1 编

码率的校验矩阵(以下称为“第 1 校验矩阵”)的列权重以下的校验矩阵(参照图 1 及图 5)。换言之,对于成为电路共用源的第 1 编码率的校验矩阵,使用列权重为最大的校验矩阵。

[0170] 列权重表示存在于校验矩阵中的各列上的元素“1”的最大个数。例如,图 1 所示的第 1 校验矩阵的列权重为 4,图 5 所示的第 2 校验矩阵的列权重也为 4,第 2 校验矩阵的列权重为第 1 校验矩阵的列权重以下。

[0171] 下面,以当作为成为电路共用源的校验矩阵使用第 1 校验矩阵时,进行使用第 2 校验矩阵的解码的情况为例进行说明。即,下面,对使用与第 1 校验矩阵对应的解码器 100 作为基本结构,既实现电路共用化,又进行与第 2 校验矩阵对应的解码的情况进行说明。

[0172] 为了实现电路的共用,与对应于图 1 的第 1 校验矩阵同样地,与图 5 的第 2 校验矩阵对应的解码算法将使第 2 校验矩阵的任意列组合所得的部分矩阵作为处理单位来进行解码处理。以下,将第 1 校验矩阵的部分矩阵称为第 1 部分矩阵,将第 2 校验矩阵的部分矩阵称为第 2 部分矩阵。

[0173] 如上所述,事前值生成单元 150 必须为部分矩阵的列数个。因此,本实施方式中,为了抑制电路规模的增大并对应于多个编码率进行解码,将构成第 2 部分矩阵的列数设为与构成第 1 部分矩阵的列数相同数。由此,事前值生成单元 $150-r$ ($r = 0 \sim 3$) 能够对应于第 1 编码率及第 2 编码率这两者。

[0174] 图 6 是表示将图 5 的第 2 校验矩阵分割成每 4 列数所得的第 2 部分矩阵的图。如图 6 所示,从第 2 校验矩阵中选择列编号 $\{0, 1, 2, 3\}$ 来形成第 2 部分矩阵 # 2-0。另外,从第 2 校验矩阵中选择列编号 $\{4, 5, 6, 7\}$ 来形成第 2 部分矩阵 # 2-1。另外,从第 2 校验矩阵中选择列编号 $\{8, 9, 10, 11\}$ 来形成第 2 部分矩阵 # 2-2。另外,从第 2 校验矩阵中选择列编号 $\{12, 13, 14, 15\}$ 来形成第 2 部分矩阵 # 2-3。这样,当第 2 部分矩阵的列数与共用源的第 1 部分矩阵的列数相同时,事前值生成单元 $150-r$ ($r = 0 \sim 3$) 能够对应于第 1 编码率及第 2 编码率这两者。

[0175] 另外,如上所述,外部值更新单元 $170-k$ ($k = 0 \sim 7$) 需要部分矩阵的行权重数个比较器。

[0176] 但是,如图 6 所示,在第 2 部分矩阵 # 2-0 的第 0 行的第 0 列、第 1 列、第 2 列、第 3 列中,分别配置有具有行权重的子矩阵即非零元素“10”、“14”、“18”、“21”。因此,第 1 部分矩阵的行权重为 2,与此相对,第 2 部分矩阵 # 2-0 的第 0 行的行权重为 4。

[0177] 因此,为了对应于第 1 部分矩阵及第 2 部分矩阵这两者,外部值更新单元 $170-k$ ($k = 0 \sim 7$) 必须具有 4 个比较器。但是,比较器的数量越增加,电路规模越增大。

[0178] 因此,本实施方式中,以对外部值更新单元 $170-k$ ($k = 0 \sim 7$) 的输入数与使用第 1 部分矩阵的情况相同数或其以下的方式,在第 2 部分矩阵使具有行权重的子矩阵(即非零元素)分散。具体而言,扩展第 2 校验矩阵的行数,将第 2 部分矩阵中的行权重大的行的非零元素配置到扩展行中,以使行权重分散。

[0179] 例如,使第 2 部分矩阵 # 2-0 的第 0 行的第 0 列的非零元素“10”与第 2 部分矩阵 # 2-0 的第 0 行的第 1 列的非零元素“14”分散到不同的行中。此时,如图 7 所示,将第 0 行分散成第 0-0 行与第 0-1 行,以将第 0 行的第 0 列的非零元素“10”包含于第 0-0 行中的方式进行配置,另外,以将第 0 行的第 1 列的非零元素“14”包含于第 0-1 行中的方式进行配置。

[0180] 这样,通过扩展第 2 校验矩阵的行数,并将第 2 部分矩阵中的行权重大的行的非零元素配置到扩展行中,从而分散行权重。图 7 中表示第 2 校验矩阵的第 0 行的第 0 列的非零元素“10”与第 1 列的非零元素“14”分别被分散成包含于第 0-0 行、第 0-1 行中的例子。

[0181] 以下,将扩展第 2 校验矩阵的行数,并将第 2 部分矩阵中的行权重大的行的非零元素配置到扩展行中所得的校验矩阵称为分散校验矩阵。此外,分散校验矩阵的行数设为成为共用源的第 1 校验矩阵的行数以下。

[0182] 图 8 表示从分散后的校验矩阵(分散校验矩阵)选择解码对象的列后的部分矩阵。通过分散第 2 校验矩阵而生成分散校验矩阵,从分散校验矩阵中选择解码对象的列数个列而生成与第 2 编码率对应的部分矩阵(以下称为“分散部分矩阵”),从而能够减小解码对象的部分矩阵的行权重。

[0183] 例如,分散后的分散部分矩阵 # 2-0 的行权重为 2,能够成为比分散前的第 2 部分矩阵的行权重 4 小。

[0184] 此外,如上所述,本实施方式中,将与第 1 编码率及第 2 编码率分别对应的第 1 校验矩阵及第 2 校验矩阵中的列权重为最大的第 1 校验矩阵用于成为电路共用源的校验矩阵(参照图 1)。即,第 1 部分矩阵(共用源的第 1 校验矩阵的部分矩阵)的某列中所含的非零元素的数量为第 2 部分矩阵的某列中所含的非零元素的数量以上。由此,在扩展共用对象的第 2 校验矩阵(参照图 5)的行数以分散行权重的情况下,能够将成为解码对象的分散部分矩阵的行权重设为共用源的第 1 部分矩阵的行权重以下。

[0185] 图 9 是表示与第 2 校验矩阵对应的解码处理的流程的图。此外,在图 9 中,在与图 3 相同的处理步骤中,本解码算法使用分散部分矩阵取代第 1 部分矩阵来进行同样的处理。

[0186] [13] 阶层结合处理 :ST201

[0187] 本解码算法中,为了作为与第 2 校验矩阵对应的解码算法,在 ST108 的后级追加阶层结合处理。本解码算法中,使矩阵元素分散,以使图 5 所示的校验矩阵成为图 7 所示的校验矩阵。因而,为了与第 2 校验矩阵对应,必须求出与使用成为源的图 5 的校验矩阵进行解码时所得的对数外部值比相同的值。为了由图 7 的分散校验矩阵求出上述相同值的对数外部值比,需要选择与分散后的两个行对应的对数外部值比中较小的对数外部值比并输出的处理。将其定义为阶层结合处理,以下的式表示处理内容。

$$[0188] \quad \alpha_{\min}(i, m, z) = \min(\alpha_{\min}(i, a, z), \alpha_{\min}(i, b, z)) \cdots \cdots (13-1)$$

$$[0189] \quad \alpha_{2nd}(i, m, z) = \min_2(\alpha_{\min}(i, a, z), \alpha_{2nd}(i, a, z), \alpha_{\min}(i, b, z), \alpha_{2nd}(i, b, z)) \cdots \cdots (13-2)$$

$$[0190] \quad \alpha_{ind}(i, m, z) = \minind(\alpha_{ind}(i, a, z), \alpha_{ind}(i, b, z)) \cdots \cdots (13-3)$$

$$[0191] \quad \alpha_{sig}(i, m, z) = \alpha_{sig}(i, a, z) \times \alpha_{sig}(i, b, z) \cdots \cdots (13-4)$$

[0192] 其中, a、b 表示通过分解第 m 行所得的扩展行的行编号。

[0193] 另外, $\minind(\cdot)$ 是在通过 $\min$ 函数选择 $\alpha_{\min}(i, a, z)$ 作为最小值的情况下返回 $\alpha_{ind}(i, a, z)$, 而在除此以外的情况下返回 $\alpha_{ind}(i, b, z)$ 的函数。

[0194] 上述阶层结合处理是在最大更新次数 itmax 的最后的解码顺序 J 中的列处理及行处理结束后(ST108 :否(N0)) 进行。

[0195] 以上,说明了与第 2 校验矩阵对应的解码算法。

[0196] 接下来,说明能够与第 1 编码率及第 2 编码率对应的解码器的结构。

[0197] 图 10 是表示本实施方式的解码器的主要部分结构的方块图。此外,在图 10 中,对于与图 4 相同的结构部分附上与图 4 相同的标号并省略说明。图 10 的解码器 100A 相对于作为基本结构的图 4 的解码器 100,取代列处理行处理运算单元 120 而具有列处理行处理运算单元 120A。此外,列处理行处理运算单元 120A 是包含列处理行处理运算单元 120 及阶层结合单元 190 的结构。

[0198] 列处理行处理运算单元 120A 具有外部值存储单元 130A-0 ~ 130A-7、行选择单元 140A-0 ~ 140A-3、事前值生成单元 150A-0 ~ 150A-3、列选择单元 160A、外部值更新单元 170A-0 ~ 170A-7 及阶层结合单元 190。

[0199] 行选择单元 140A-0 ~ 140A-3 与行选择单元 140-0 ~ 140-3 的不同之处在于:根据所输入的设定编码率的信息,将所用的部分矩阵设定为第 1 部分矩阵或分散部分矩阵中的任一者。若设定部分矩阵,则以后,行选择单元 140A-0 ~ 140A-3 与行选择单元 140-0 ~ 140-3 同样地,选择与所用的部分矩阵的非零元素对应的对数外部值比并输出。这样,行选择单元 140A-0 ~ 140A-3 对应于不同的编码率来进行行处理。

[0200] 事前值生成单元 150A-0 ~ 150A-3 与事前值生成单元 150-0 ~ 150-3 的不同之处在于:根据所输入的设定编码率的信息,将所用的部分矩阵设定为第 1 部分矩阵或分散部分矩阵中的任一者。若设定部分矩阵,则以后,事前值生成单元 150A-0 ~ 150A-3 与事前值生成单元 150-0 ~ 150-3 同样地,根据所用的部分矩阵的循环移位量来进行对数事前值比的移位(重新排列)并输出。这样,行选择单元 140A-0 ~ 140A-3 对应于不同的编码率来输出对数外部值比。

[0201] 此外,与事前值生成单元 150-0 ~ 150-3 同样地,事前值生成单元 150A-0 ~ 150A-3 具有加法器,使用加法器计算对数事前值比。如上所述,第 1 部分矩阵及分散部分矩阵由同数列构成,因此解码器 100A 只要具有第 1 部分矩阵的列数个的事前值生成单元即可。因此,由加法器构成的事前值生成单元 150A-0 ~ 150A-3 能够对于不同的编码率而共用。

[0202] 列选择单元 160A 与列选择单元 160 的不同之处在于:根据所输入的设定编码率的信息,将所用的部分矩阵设定为第 1 部分矩阵或分散部分矩阵中的任一者。若设定部分矩阵,则以后,列选择单元 160A 与列选择单元 160 同样地,选择与所用的部分矩阵的非零元素对应的对数事前值比并输出。这样,列选择单元 160A 对于不同的编码率进行列处理。

[0203] 外部值更新单元 170A-0 ~ 170A-7 与外部值更新单元 170-0 ~ 170-7 的不同之处在于:根据所输入的设定编码率的信息,将所用的部分矩阵设定为第 1 部分矩阵或分散部分矩阵中的任一者。若设定部分矩阵,则以后,外部值更新单元 170A-0 ~ 170A-7 与外部值更新单元 170-0 ~ 170-7 同样地,使用对数事前值比来更新与各行对应的对数外部值比。这样,外部值更新单元 170A-0 ~ 170A-7 对应于不同的编码率来更新对数外部值比。

[0204] 此外,分散后的校验矩阵(分散校验矩阵)的行权重为共用源的校验矩阵(第 1 校验矩阵)的行权重以下。因而,对外部值更新单元 170A-0 ~ 170A-7 的输入数为共用源的校验矩阵的行权重的数以下,因此外部值更新单元 170A-0 ~ 170A-7 能够对于不同的编码率来进行行处理。即,构成外部值更新单元 170A-0 ~ 170A-7 的比较器能够对于不同的编码率而共用。

[0205] 从外部值更新单元 170A-0 ~ 170A-7 输出的对数外部值比被输入阶层结合单元

190。阶层结合单元 190 使用所输入的对数外部值比来进行式(13-1)~(13-4)所示的运算。阶层结合单元 190 通过进行式(13-1)~(13-4)所示的运算,求出与分散前的第 2 校验矩阵对应的对数外部值比。

[0206] 此外,如式(13-1)~(13-4)所示,通过比较来求出与第 2 校验矩阵对应的对数外部值比。因此,阶层结合单元 190 在内部具有比较器。

[0207] 阶层结合单元 190 将与分散前的第 2 校验矩阵的行对应的对数外部值比作为与分散后的分散校验矩阵的行对应的对数外部值比来进行分配,并将分配后的对数外部值比输出到外部值存储单元 130A-0~130A-7。

[0208] 外部值存储单元 130A-0~130A-7 存储从阶层结合单元 190 输出的对数外部值比。

[0209] 如上所述,在本实施方式中,解码器 100A 是基于与设定编码率对应的校验矩阵,对通过从多个编码率中设定的设定编码率进行了 LPDC 编码的编码比特进行解码的解码器。并且,列处理行处理运算单元 120A 使用似然及与对应于设定编码率的校验矩阵相应的部分矩阵,反复进行列处理及行处理以计算算软判定值。具体而言,当设定编码率为第 1 编码率时,列处理行处理运算单元 120A 使用根据列处理中的解码对象的列数从与第 1 编码率对应的第 1 校验矩阵中选择并组合而成的第 1 部分矩阵。这里,列处理中的解码对象的列数是指 1 以上的列,是考虑电路规模来设定。另一方面,当设定编码率为第 2 编码率时,列处理行处理运算单元 120A 使用从与对应于第 2 编码率的第 2 校验矩阵相应的分散校验矩阵中选择与构成第 1 部分矩阵的列数相同数的列而组合而成的分散部分矩阵。此处,第 2 编码率是比第 1 编码率大的编码率。

[0210] 这里,分散校验矩阵是扩展第 2 校验矩阵的行数,并将第 2 校验矩阵中的行权重大的行的元素分散配置到该行与扩展行所得的矩阵。此时,列处理行处理运算单元 120A 使用行权重为第 1 部分矩阵的行权重以下的分散部分矩阵。

[0211] 由此,不论设定编码率如何,向外部值更新单元 170A-k ($k = 0 \sim 7$) 输入的对数事前值比的数量均为第 1 部分矩阵的行权重数以下。其结果,能够在不同的编码率(第 1 编码率及第 2 编码率)下共用构成外部值更新单元 170A-k ($k = 0 \sim 7$) 的比较器。

[0212] 另外,本实施方式中,将第 1 校验矩阵及分散校验矩阵分割成第 1 部分矩阵及分散部分矩阵,将分割所得的第 1 部分矩阵及分散部分矩阵作为处理单位来进行列处理。这里,分散部分矩阵的列数与第 1 部分矩阵的列数相同数。由此,不论设定编码率如何,事前值生成单元 150A-m ($m = 0 \sim 3$) 均需要构成第 1 部分矩阵及分散部分矩阵的列数个。即,能够在不同的编码率下共用使用加法器构成的事前值生成单元 150A-m ($m = 0 \sim 3$)。

[0213] 另外,向外部值存储单元 130A-k ($k = 0 \sim 7$) 输入与构成第 1 校验矩阵的行数相同数的对数事前值比。因此,通过将分散校验矩阵的行数设为第 1 校验矩阵的行数以下,能够在不同的编码率下共用外部值存储单元 130A-k ($k = 0 \sim 7$)。

[0214] 这样,根据本实施方式,能够共用对数外部值比的存储装置(外部值存储单元)、用于运算对数事前值比的加法器(事前值生成单元)、以及用于更新对数外部值比的比较器(外部值更新单元)。

[0215] 此外,图 10 所示的解码器 100A 采用在外部值更新单元 170A-k ($k = 0 \sim 7$) 的后级包含阶层结合单元 190 的结构。在无线 LAN 系统或移动电话系统中的一对一的数据通信

模式的情况下,接收机在解码接收数据后,必须在规定时间内向发送机回复确认响应。因而,为了在规定时间内回复确认响应,解码所需的处理时间必须至少短于规定时间。因此,在外部值更新单元 170A-k ($k = 0 \sim 7$)的后级设有阶层结合单元 190 的图 10 所示的解码器 100A 的结构适合于缩短解码处理时间的用途。

[0216] 另一方面,在广播系统或一对多的组播数据通信模式中,接收机无须对发送机回复确认响应。即,不存在对解码所需的处理延迟的限制。在此种情况下,也可以不独立设置阶层结合单元 190,而是如图 11 所示的解码器 100B 那样,在列处理行处理运算单元 120B 中,外部值更新单元 170B-k ($k = 0 \sim 7$)除了更新对数外部值比以外,还进行阶层结合处理。

[0217] 如上所述,使用比较器构成外部值更新单元及阶层结合单元。因此,能够将外部值更新单元的比较器也用于阶层结合处理。这样,通过实现构成阶层结合单元的比较器与构成外部值更新单元的比较器的共用化,能够削减解码器 100B 的电路规模。

[0218] 以下,对于采用图 10 及图 11 所示的结构的解码器 100A、100B 中的解码处理,使用图 12 及图 13 所示的时间图来进行说明。

[0219] 图 12 是表示图 10 所示的解码器 100A 中的解码处理的时间图的图。图 12 作为一例示出了外部值更新单元 170A-0、170A-1 中的处理。

[0220] 外部值更新单元 170A-0、170A-1 按照图 8 所示的解码顺序的顺序进行外部值更新。在第 3 个解码顺序中的外部值更新完成的时刻,等同于对图 7 所示的分散校验矩阵整体的第 i 次解码处理完成。在外部值更新完成的同时,阶层结合单元 190 使用外部值更新单元 170A-0、170A-1 所输出的对数外部值比进行阶层结合处理。阶层结合单元 190 将阶层结合后的对数外部值比反馈给外部值存储单元 130A-0、130A-2。当反馈完成时,就开始第 $i + 1$ 次的解码处理。

[0221] 图 10 所示的解码器 100A 独立于外部值更新单元 170A-1 $\sim$ 170A-7 而具有阶层结合单元 190。因而,在外部值更新单元 170A-1 $\sim$ 170A-7 完成外部值更新处理的同时,阶层结合单元 190 能够进行阶层结合处理。由此,能够使反复解码处理中的 1 次解码处理所需要的处理循环数达到最小。如图 12 所示,当独立具有阶层结合单元 190 时,1 次解码处理中的循环数为 4。

[0222] 另一方面,图 13 是表示图 11 所示的解码器 100B 中的解码处理的时间图的图。为了与图 12 进行对比,图 13 示出外部值更新单元 170B-0、170B-1 中的处理。此外,对外部值更新单元 170B-0、170B-1 计算针对分散校验矩阵中的分散有行权重的行的对数外部值比的情况进行说明。

[0223] 外部值更新单元 170B-0、170B-1 按照图 8 所示的解码顺序的顺序进行外部值更新。当第 3 个解码顺序中的外部值更新完成,且第 i 次解码处理完成时,在下个处理循环中进行阶层结合处理。外部值更新单元 170B-0、170B-1 在阶层结合处理完成时,将所得的对数外部值比反馈给外部值存储单元 130A-0、130A-2。当反馈完成时,就开始第 $i + 1$ 次的解码处理。

[0224] 在图 11 中,外部值更新单元 170B-1 将在第 3 个解码顺序中更新的对数外部值比输出到外部值更新单元 130A-0。而且,外部值更新单元 170B-0 使用在第 3 个解码顺序中更新的对数外部值比与从外部值更新单元 170B-1 输入的对数外部值比,进行式(13-1) $\sim$

(13-4) 所示的阶层结合处理。

[0225] 此外,在形成分散校验矩阵时,若分散有行权重的行存在多个,则计算针对该行的对数外部值的外部值更新单元 170B-p、170B-(p+1) (p 为正的偶数) 与外部值更新单元 170B-0、170B-1 同样地进行阶层结合处理。

[0226] 并且,外部值更新单元 170B-0、170B-1 将通过阶层结合获得的对数外部值比分配给与分散后的行对应的对数外部值比,各外部值更新单元 170B-0、170B-1 将分配后的对数外部值比输出到外部值存储单元 130A-0、130A-1。

[0227] 这样,图 11 所示的解码器 100B 不另行设置阶层结合单元 190,而是使用构成外部值更新单元 170B-k ($k = 0 \sim 7$) 的比较器来进行阶层结合处理。此时,如图 13 所示,1 次解码处理所需的处理循环数为 5。其结果,如图 10 所示,比起采用独立于外部值更新单元 170A-k ($k = 0 \sim 7$) 而具有阶层结合单元 190 的结构的情况,解码处理循环数增加。但是,阶层结合处理所需的比较器能够与外部值更新单元 170B-k ($k = 0 \sim 7$) 所保持的比较器共用,因此与解码器 100A 相比,能够削减解码器 100B 的电路规模。

[0228] 这样,图 10 的解码器 100A 成为适合于缩短解码处理时间的用途,例如适合于一对一的数据通信模式的结构。另一方面,图 11 的解码器 100B 尽管解码处理时间稍长,但能够进一步缩小电路规模,例如适合于广播系统或一对多的组播数据通信模式的结构。

[0229] 在 2010 年 8 月 6 日申请的特愿 2010-177538 的日本专利申请所包含的说明书、附图和说明书摘要的公开内容,全部引用于本申请。

[0230] 工业实用性

[0231] 本发明的纠错解码装置及纠错解码方法对于适用 LDPC 码的通信装置等有用。另外,还能够应用于广播接收装置或存储介质等。

行编号(m) ↓

列编号(n) →

循环移位量

零矩阵

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	-	8	-	16	-	24	-	32	-	-	-	-	-	-	-
1	1	-	9	-	17	-	-	28	33	36	-	-	-	-	-	-
2	-	4	-	12	-	20	-	29	-	37	39	-	-	-	-	-
3	-	5	-	13	-	21	25	-	-	-	40	42	-	-	-	-
4	2	-	10	-	18	-	26	-	34	-	-	43	45	-	-	-
5	3	-	11	-	-	22	-	30	-	38	-	44	-	47	-	-
6	-	6	-	14	-	23	-	31	-	-	41	-	-	48	49	-
7	-	7	-	15	19	-	27	-	35	-	-	-	46	-	50	51

图 1

列编号

→

0	1	2	3								
0	-	8	-								
1	-	9	-								
-	4	-	12								
-	5	-	13								
2	-	10	-								
3	-	11	-								
-	6	-	14								
-	7	-	15								

部分矩阵#1-0

每

列编号

→

4	5	6	7								
16	-	24	-								
17	-	-	28								
-	20	-	29								
-	21	25	-								
18	-	26	-								
-	22	-	30								
-	23	-	31								
19	-	27	-								

部分矩阵#1-1

每

列编号

→

8	9	10	12								
32	-	-	-								
33	36	-	-								
-	37	39	-								
-	-	40	-								
34	-	-	45								
-	38	-	-								
-	-	41	-								
35	-	-	46								

部分矩阵#1-2

部

列编号

→

11	13	14	15								
-	-	-	-								
-	-	-	-								
-	-	-	-								
42	-	-	-								
43	-	-	-								
44	47	-	-								
-	48	49	-								
-	-	50	51								

部分矩阵#1-3

部

解码顺序

→

0	1	2	3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51

图 2

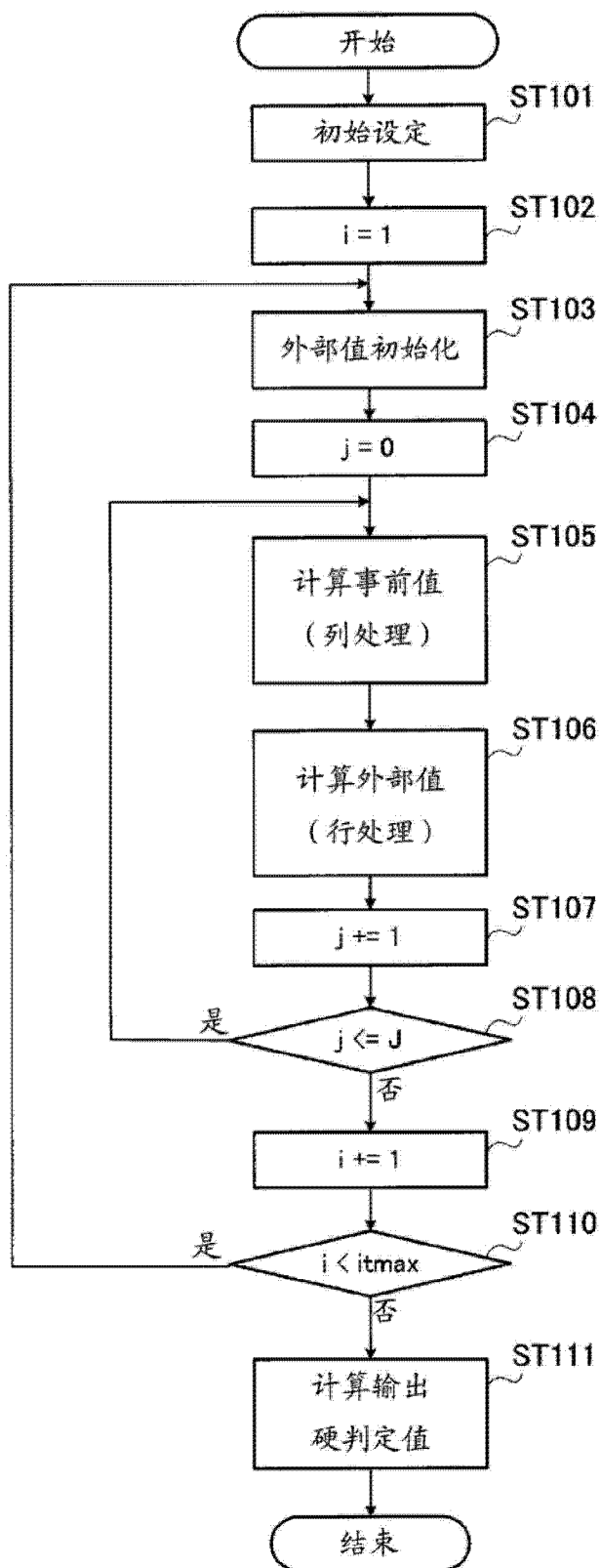


图 3

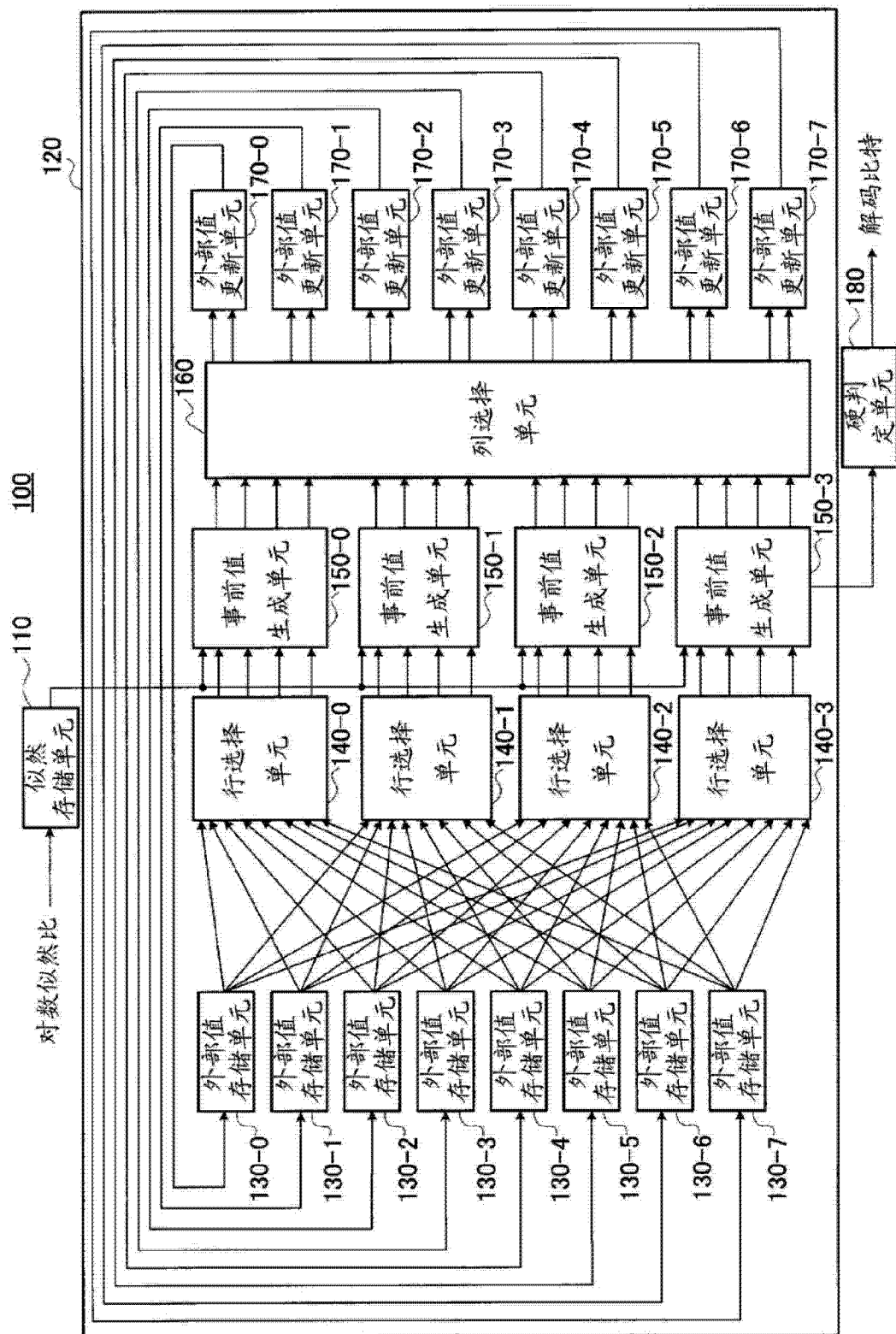
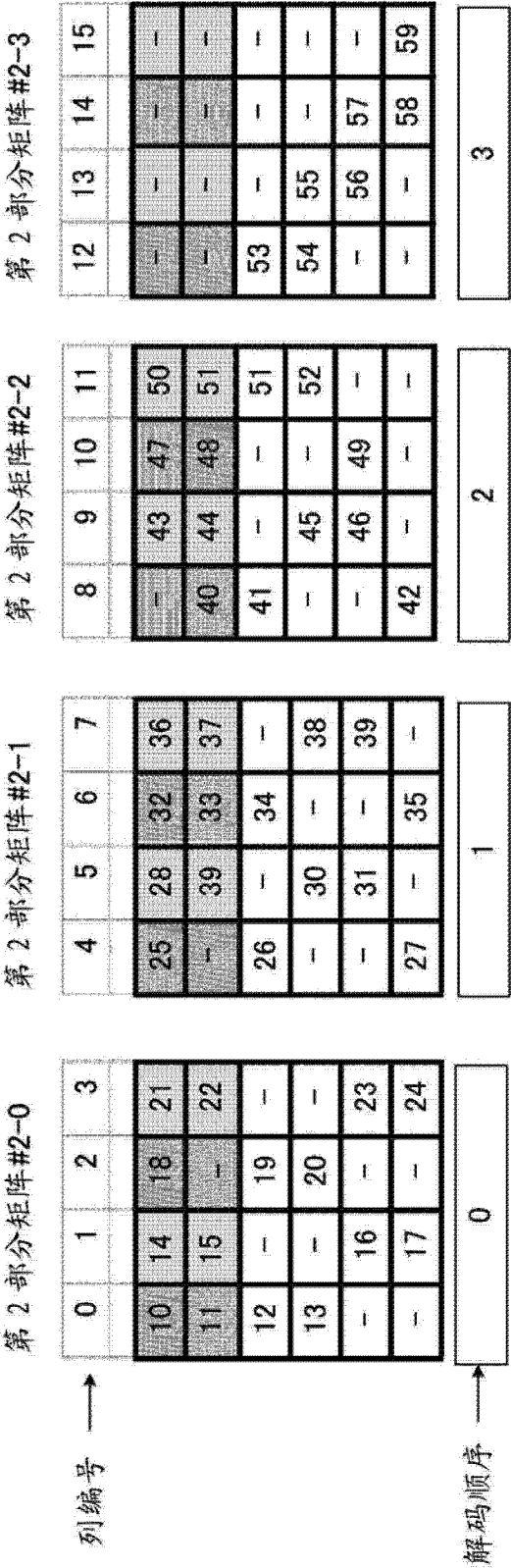


图 4

	列编号															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
行编号	0	10	14	18	21	25	28	32	36	-	43	47	-	-	-	-
1	11	15	-	22	-	29	33	37	40	44	48	50	-	-	-	-
2	12	-	19	-	-	-	34	-	41	-	-	51	53	-	-	-
3	13	-	20	-	-	30	-	38	-	45	-	52	54	55	-	-
4	-	16	-	23	-	31	-	39	-	46	49	-	-	56	57	-
5	-	17	-	24	27	-	35	-	42	-	-	-	-	-	58	59

图 5



第 2 部分矩阵#2-1

4	5	6	7
25	28	32	36
-	39	33	37
26	-	34	-
-	30	-	38
-	31	-	39
27	-	35	-

1

第 2 部分矩阵#2-2

8	9	10	11
-	43	47	50
40	44	48	51
41	-	-	51
-	45	-	52
-	46	49	-
42	-	-	-

2

第 2 部分矩阵#2-3

12	13	14	15
-	-	-	-
-	-	-	-
53	-	-	-
54	55	-	-
-	56	57	-
-	-	58	59

3

图 6

列编号	行编号															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0-0	10	-	18	-	25	-	32	-	-	-	47	-	-	-	-	-
1-0	11	-	-	-	-	-	33	-	40	-	48	-	-	-	-	-
0-1	-	14	-	21	-	28	-	36	-	43	-	-	-	-	-	-
1-1	-	15	-	22	-	29	-	37	-	44	-	50	-	-	-	-
2	12	-	19	-	26	-	34	-	41	-	-	51	53	-	-	-
3	13	-	20	-	-	30	-	38	-	45	-	52	54	55	-	-
4	-	16	-	23	-	31	-	39	-	46	49	-	-	56	57	-
5	-	17	-	24	27	-	35	-	42	-	-	-	-	-	58	59

图 7



图 8

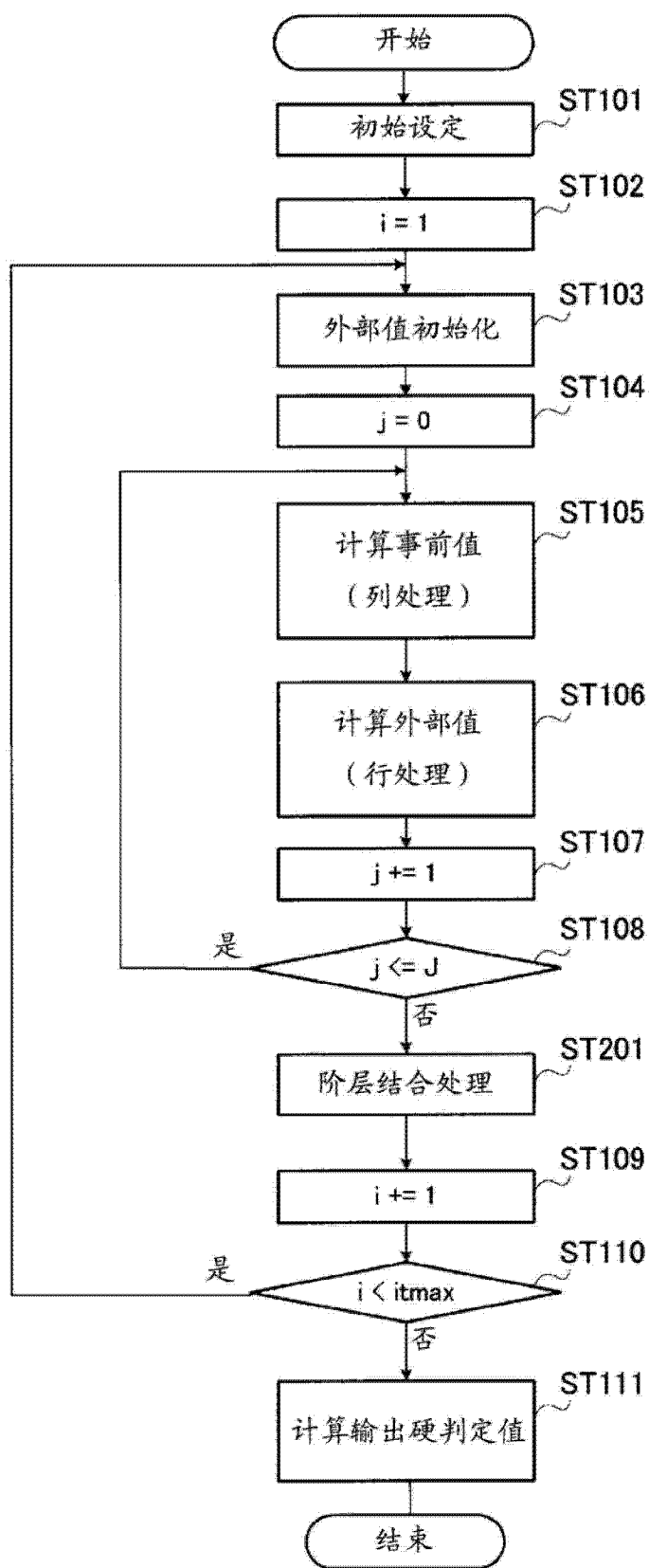


图 9

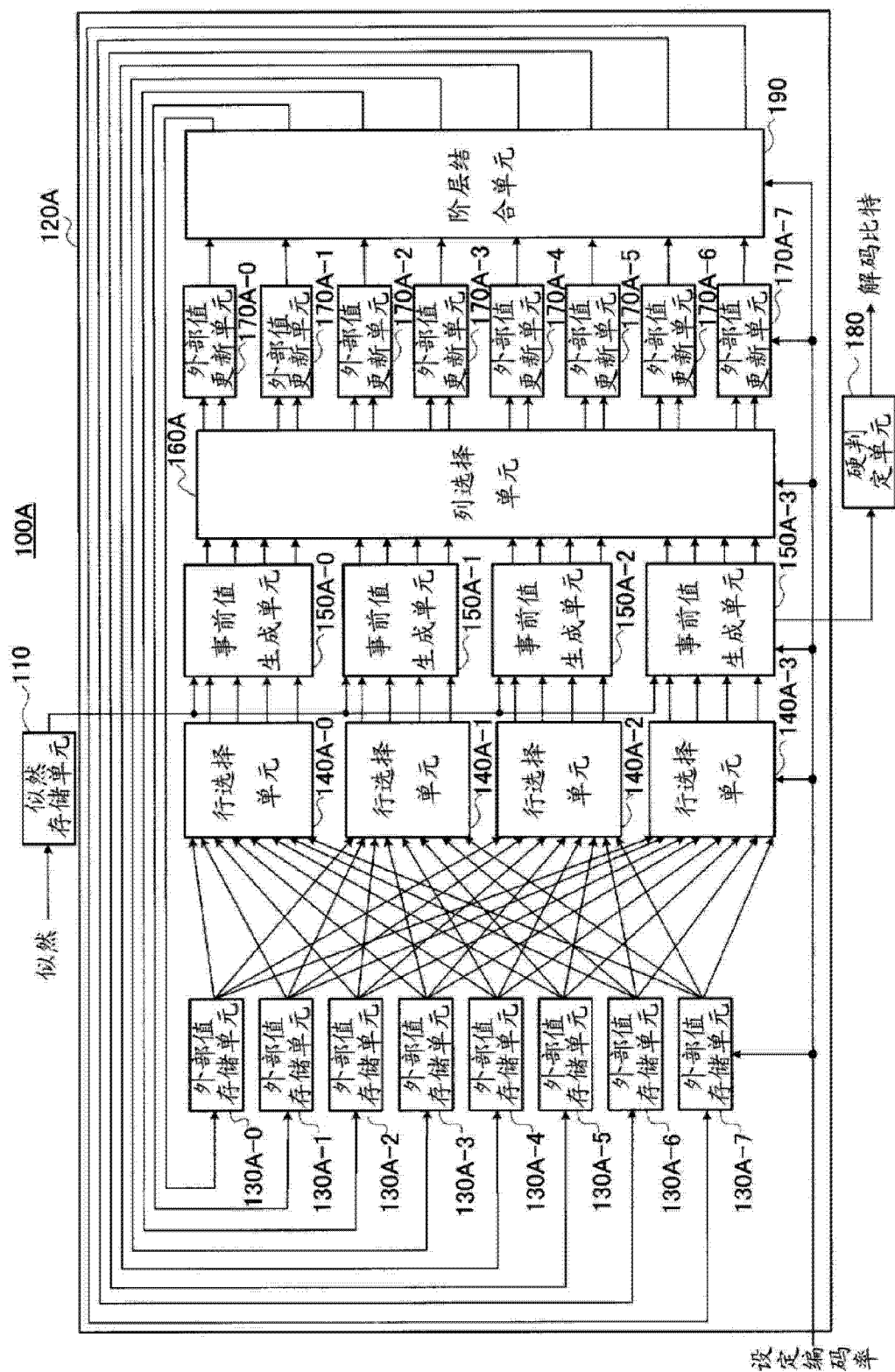


图 10

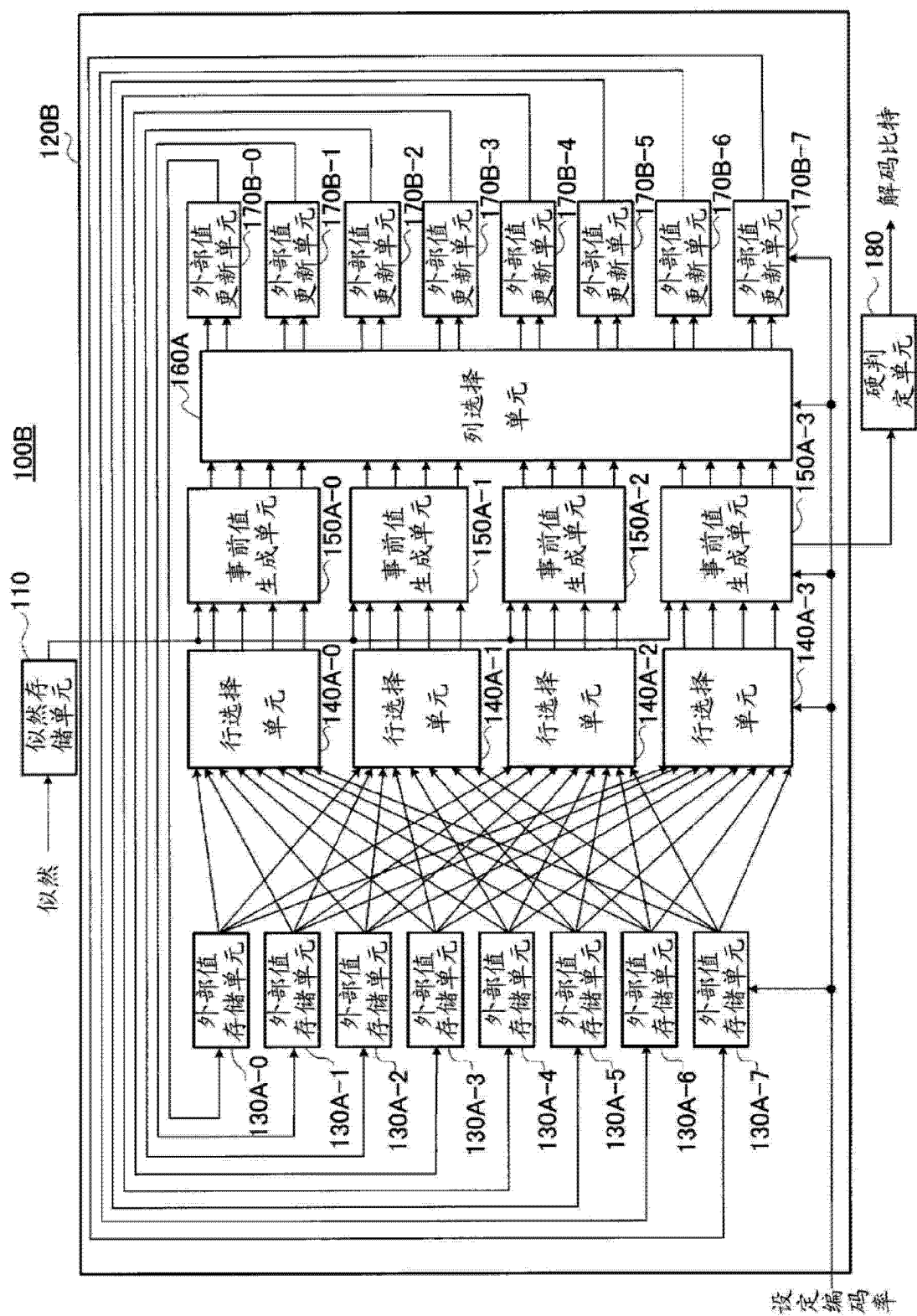


图 11

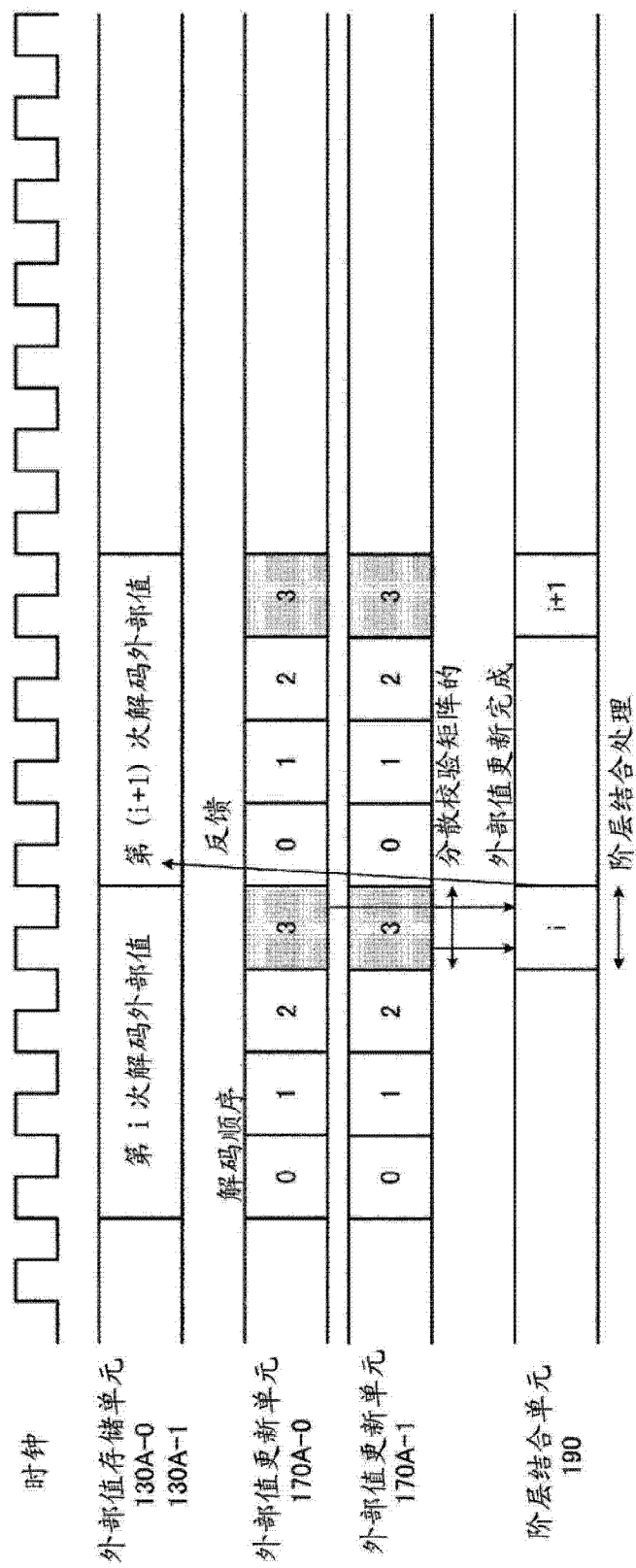


图 12

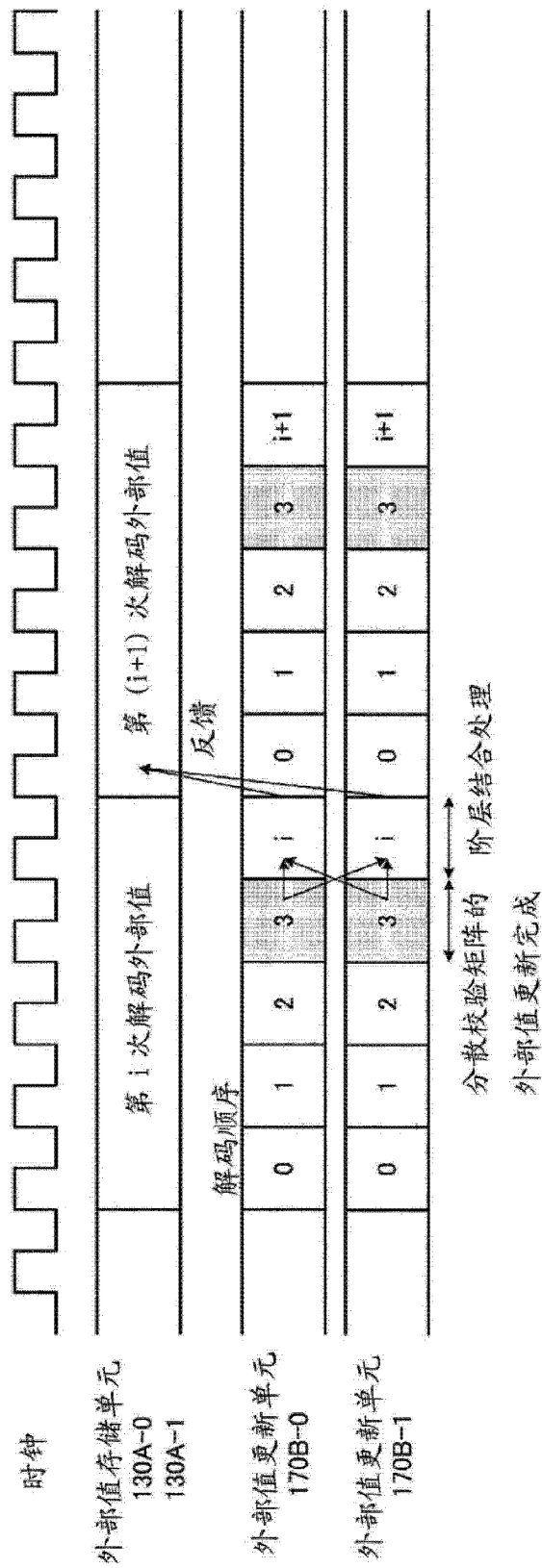


图 13

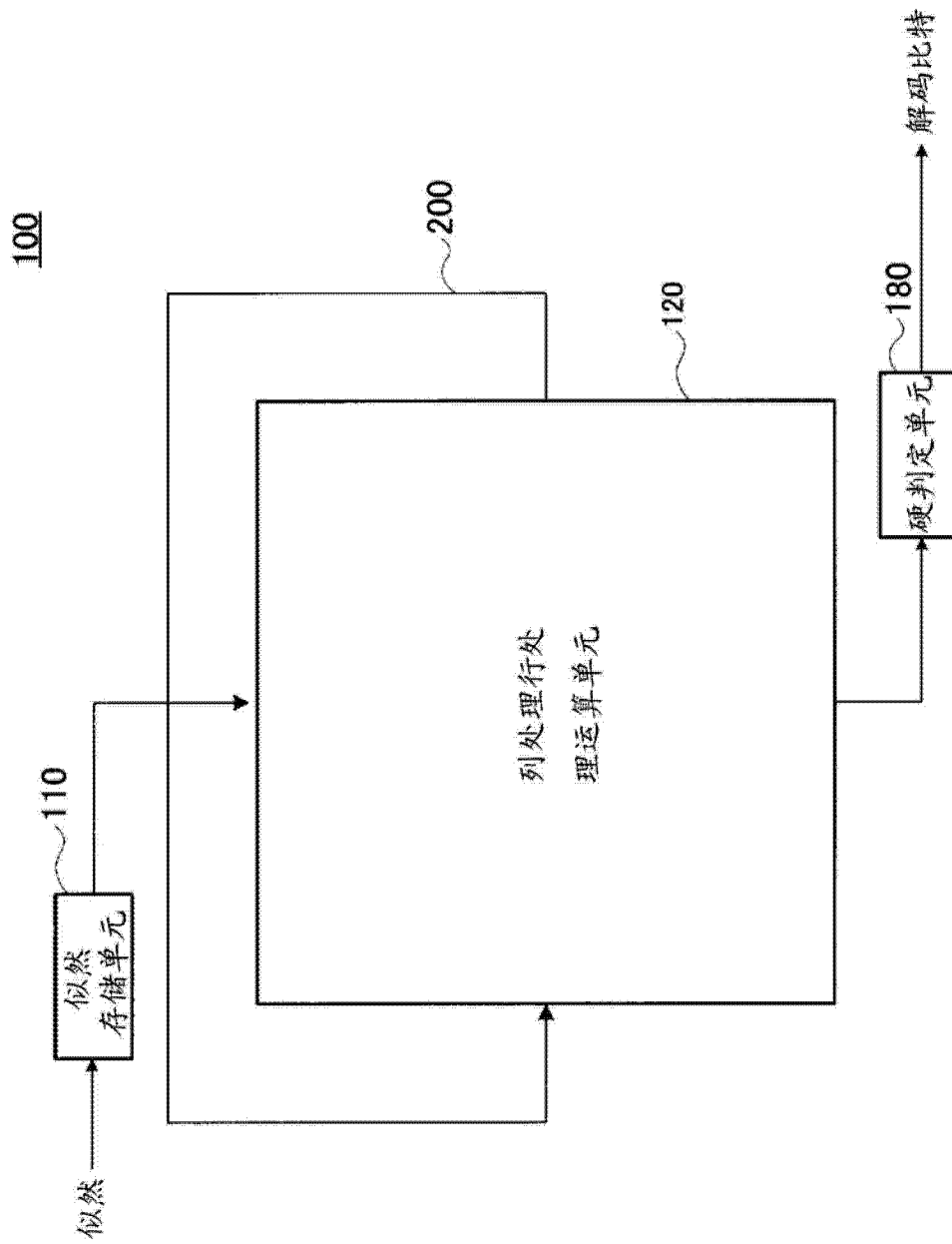


图 14

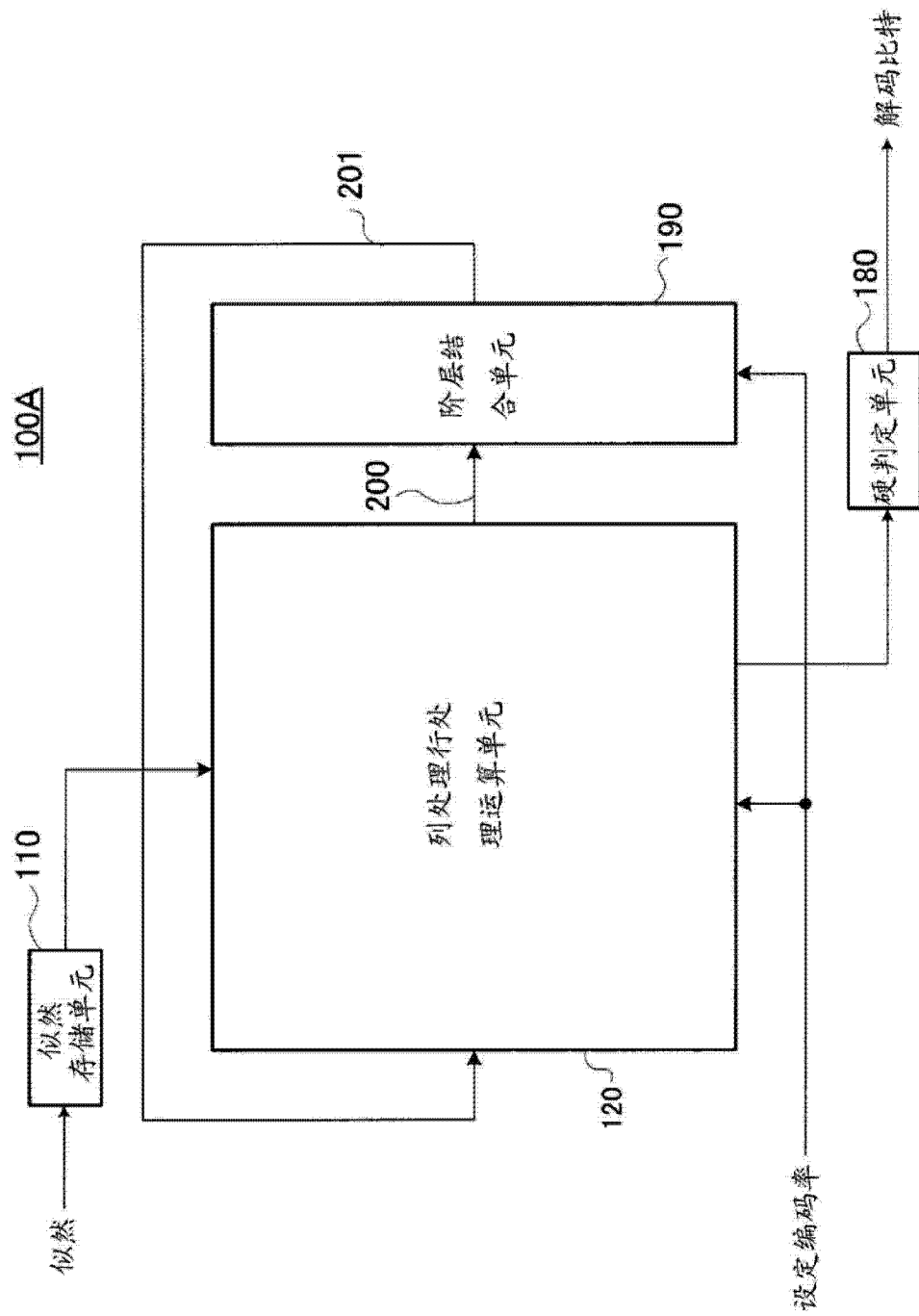


图 15