



(12) 发明专利申请

(10) 申请公布号 CN 104488208 A

(43) 申请公布日 2015.04.01

(21) 申请号 201380039087.7

代理人 王茂华 马明月

(22) 申请日 2013.07.23

(51) Int. Cl.

(30) 优先权数据

H04L 1/00(2006.01)

12305947.9 2012.08.01 EP

H03M 13/27(2006.01)

(85) PCT国际申请进入国家阶段日

2015.01.22

(86) PCT国际申请的申请数据

PCT/EP2013/065455 2013.07.23

(87) PCT国际申请的公布数据

W02014/019881 EN 2014.02.06

(71) 申请人 阿尔卡特朗讯

地址 法国布洛涅比扬古

(72) 发明人 A·迪帕 R·布瓦莱格

(74) 专利代理机构 北京市金杜律师事务所

11256

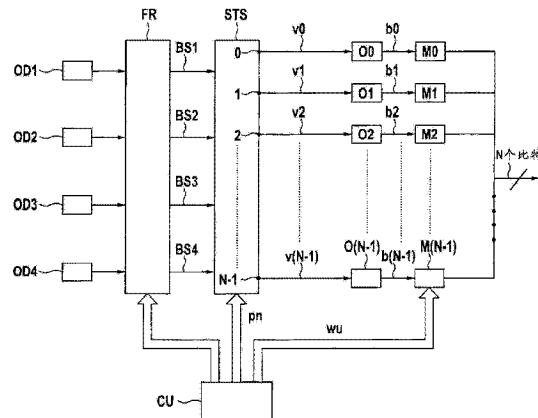
权利要求书2页 说明书9页 附图6页

(54) 发明名称

用于光线路终端的比特交织器

(57) 摘要

提出一种用于光接入网络的光线路终端的比特交织器。比特交织器包含在比特级向空间 - 时间交换机提供数据流的存储器读取器。空间 - 时间交换机在一个输入周期内从数据流读取上至 N 个比特集合。交换机在一个写入周期内将上至 N 个比特交换到上至它的提供相应输出矢量的输出端口上。数目为 N 的 OR 函数元件在写入周期内确定相应单个输出比特。数目为 N 的存储器元件在一个写入周期内将输出比特中的相应输出比特写入到它们的比特子元件中的相应比特子元件中。控制单元控制对数据流的读取并且也控制交换机对比特的交换。控制单元控制写入地址的选择。



1. 一种用于光接入网络的光线路终端的比特交织器，包括：

- 存储器读取器 (FR)，可操作用于接收和提供向相应光网络单元指配的数目为 M 的数据流 (OD1, …, OD4)；

- 空间 - 时间交换机 (STS)，所述空间 - 时间交换机 (STS)：

- 可操作用于在一个输入周期内从所述相应 M 个数据流读取上至 M 个相应比特集合 (BS1, …, BS4)；

- 并且还可操作用于在一个写入周期内将所述数据流 (OD1, …, OD4) 的上至 N 个比特交换到提供相应输出矢量 (v0, …, v(N-1)) 的上至 N 个相应输出端口上；

-N 个或函数元件 (01, …, 0(N-1))，每个或函数元件可操作用于在所述一个写入周期内基于所述相应输出矢量 (v0, …, v(N-1)) 中的一个输出矢量来确定相应单个输出比特 (b0, …, b(N-1))；

-N 个存储器元件 (M0, …, M(N-1))，每个存储器元件包括具有相应子元件地址的至少 L 个比特子元件，并且每个存储器元件可操作用于在所述一个写入周期内将所述相应单个输出比特 (b0, …, b(N-1)) 中的相应单个输出比特写入到它的比特子元件中的相应比特子元件中；

所述比特交织器还包括控制单元 (CU)，所述控制单元 (CU) 可操作用于控制：

- 所述空间 - 时间交换机 (STS) 对所述 M 个数据流 (M0, …, M(N-1)) 的读取；

- 所述空间 - 时间交换机 (STS) 对所述比特到所述输出端口上的交换；

- 以及对所述比特子元件的所述写入地址的选择；

其中所述控制单元 (CU) 可操作用于根据以下各项选择所述输出端口和所述写入地址：

- 预定义的数据速率；

- 预定义的偏移值；

- 在比特帧内的比特的最大数目 B，以及

- 数目 N；

- 以及数目 L。

2. 根据权利要求 1 所述的比特交织器，

其中所述控制单元 (CU) 可操作用于控制所述空间 - 时间交换机 (STS)，使得以预定义的数据速率读取所述数据流 (OD1, …, OD4)；

并且其中所述控制单元 (CU) 可操作用于在所述一个写入周期内将所述比特子元件的所述写入地址选择为相同地址值。

3. 根据权利要求 1 所述的比特交织器，

其中从所述相应数据流读取的所述相应比特集合 (BS1, …, BS4) 中的每个比特集合包含 K 个比特。

4. 根据权利要求 1 所述的比特交织器，

其中所述控制单元 (CU) 还可操作用于控制用于在一个读取周期内从所述 N 个存储器元件读取相应 N 个比特的相应读取地址。

5. 根据权利要求 1 所述的比特交织器，

其中所述控制单元 (CU) 可操作用于在所述一个读取周期内将所述读取地址选择为相

同地址值。

6. 一种用于光接入网络的光线路终端，

包括根据权利要求 1 至 6 中的一项所述的比特交织器 (BI)。

7. 一种用于光接入网络的比特交织的方法，包括：

- 接收和提供向相应光网络单元指配的数目为 M 的数据流 (OD1, ..., OD4)，

- 使用空间 - 时间交换机 (STS) 在一个输入周期内从所述相应 M 个数据流读取上至 M 个相应比特集合 (BS1, ..., BS4)，

- 使用所述空间 - 时间交换机 (STS)，在一个写入周期内将所述数据流 (OD1, ..., OD4) 的上至 N 个比特交换到提供相应输出矢量的上至 N 个相应输出端口上，

- 在所述一个写入周期内基于所述相应输出矢量 (v0, ..., v(N-1)) 中的一个输出矢量来确定相应单个输出比特 (b0, ..., b(N-1))，

- 在所述一个写入周期内将所述相应单个输出比特 (b0, ..., b(N-1)) 中的相应单个输出比特写入到相应 N 个存储器元件 (M0, ..., M(N-1)) 的相应比特子元件中，其中所述相应比特子元件具有相应子元件地址，

所述方法还包括控制：

- 所述空间 - 时间交换机 (STS) 对所述 M 个数据流 (M0, ..., M(N-1)) 的读取，

- 所述空间 - 时间交换机 (STS) 对所述比特到所述输出端口上的交换，

- 以及对所述比特子元件的所述写入地址的选择。

用于光线路终端的比特交织器

技术领域

[0001] 本发明涉及电信领域、具体地涉及一种比特交织器和一种用于光接入网络的光线路终端的比特交织方法。

背景技术

[0002] 光接入网络如今是一种用于从光线路终端向多个光网络单元发送数据的主要解决方案。光线路终端位于中央局端，其中光线路终端充当在光接入网络与用于数据传输的核心网络之间的接口。光线路终端从核心网络接收数据并且在下游方向上朝着光网络单元(ONU)发送这一数据，客户可以将它们的设备连接到这些ONU用于数据传输。向远程节点连接到的馈送器光纤中发送光线路终端生成的下游信号。这一远程节点将下游信号拆分到不同光网络单元连接到的不同光支路上。

[0003] 一种用于发送向不同光网络单元指配的不同数据流的解决方案是在下游信号内为每个光网络单元分配一个或者多个时间隙，光线路终端在该一个或者多个时间隙内放置数据流的向相应光网络单元指配的多个数据比特。在这样的解决方案中，相应光网络单元需要以某个数据速率在这样的时间隙内接收数据，而在向其它光网络单元指配的其它时间隙期间，相应光网络单元无需以任何数据速率接收数据。这意味着光网络单元必须能够在指明的时间隙内以比从光线路终端向相应光网络单元发送指配的数据流的总平均数据速率更高的数据速率接收数据。

[0004] 所谓比特交织无源光网络(BIPON)的比特交织协议提供一种备选解决方案，在该备选解决方案内，光网络单元可以恒定数据速率从光线路终端接收数据。在这样的BIPON中，在全局帧内交织向不同ONU指配的不同数据流的比特数据，从而用于每个ONU的每个数据流的所得数据速率具有相应恒定值。全局帧可以例如包含上至 $8*19,200$ 字节 = 153,600字节，其等于1,228,800比特。在全局帧内相互等距离地放置一个具体数据流的比特，这产生用于这一数据流的所得恒定数据速率。全局帧然后被用于进行中的数据传输的更多连续全局帧跟随。

[0005] 假设全局帧的给定的持续时间，用于具体数据流的数据速率由这一数据流的比特被间隔的比特位置数目定义。通过在全局帧内放置具有不同相应等距离间隔的不同数据流的比特，实现不同数据速率。BIPON的优点是ONU需要不以全局帧提供的总数据速率、但是以由全局帧的持续时间和这一ONU的这一具体数据流的比特被放置在全局帧以内的速率定义的更低数据速率接收数据。这允许ONU以比BIPON的连续全局帧实现的最大数据速率更低的数据速率操作。

[0006] 进而进一步，通过改变具体数据流的数据比特被放置在全局帧以内的速率，对于关联光网络单元因此实现改变的数据速率。

[0007] 在BIPON内，光线路终端因此必须执行对向不同光网络单元指配的不同数据流的接收并且也必须执行不同数据流的比特向连续全局帧中的恰当比特交织，从而对于相应不同光网络单元实现不同预定数据速率。

发明内容

- [0008] 本发明的目的是改进不同数据流在光线路终端的比特交织的已知方法。
- [0009] 提出一种用于光接入网络的光线路终端的比特交织器。光接入网络优选地是BIPON类型的无源光网络。
- [0010] 提出的比特交织器包含不同子设备。
- [0011] 比特交织器包含能够接收向相应光网络单元指配的数目为M的数据流的存储器读取器。存储器读取器在比特级向空间-时间交换机提供这些数据流，该空间-时间交换机是比特交织器的又一子设备。
- [0012] 空间-时间交換机能够在一个输入周期内从相应M个数据流读取上至N个相应比特集合。
- [0013] 另外，空间-时间交換机能够在一个写入周期内将数据流的N个比特交换到上至N个相应输出端口上。相应输出端口提供相应输出矢量。空间-时间交換机执行相应比特集合在比特级的交換。
- [0014] 比特交织器还包含能够基于相应输出矢量在写入周期内确定相应单个输出比特的N个或(OR)函数元件。因此，OR函数元件通过按照布尔OR函数组合一个输出矢量的不同比特来从一个输出矢量确定所得单个输出比特。
- [0015] 比特交织器还包含N个存储器元件，这些存储器元件又各自包含至少L个比特子元件，其中这些比特子元件具有相应子元件地址。N个存储器元件各自可操作于在一个写入周期内将相应单个输出比特中的相应单个输出比特写入到它们的比特子元件中的相应比特子元件中。
- [0016] 比特交织器还包含能够控制空间-时间交換机对M个数据流的读取的控制单元。另外，控制单元能够控制空间-时间交換机将比特交换到空间-时间交換机的输出端口上。进而另外，控制单元能够控制写入地址的选择。写入地址是所得单个输出比特被写入到相应比特子元件中的那些地址。
- [0017] 为了领会提出的比特交织器的优点，必须考虑以下方面。
- [0018] 在必须在光线路终端将不同数据流的比特交织到比特交织协议的全局帧中时，一种简单解决方案将是根据希望的比特交织模式将不同数据流的不同比特写入到一个大存储器中。接着，将在已经将所有比特写入到全局存储器中之后从一个大存储器读取出全局帧的所有不同比特。这样做的结果将是最后一个比特写入到这一大存储器中之前不能从这一存储器读取出全局帧的第一个比特。因此，向保持整个全局帧的单个大存储器中写入的所有数据流将经历与全局帧的持续时间相等的延时。
- [0019] 提出的比特交织器具有可以将不同M个数据流的少于B个数据比特写入到N个存储器元件中、然后在已经将这些比特写入到这些N个存储器元件中之后读取出这些比特这样的优点。存储器元件的数目N小于在全局帧内存在的比特的最大数目B。例如在一个写入周期内，可以将N个比特写入到N个存储器元件中的相应存储器元件中，而在这一写入周期完成之后，可以在可以恰在写入周期已经完成之后开始的读取周期内读出这些N个比特。
- [0020] 如先前提到的那样，存储器元件的数目N小于在全局帧内存在的比特的最大数目

B。通过在连续读取周期从 N 个存储器元件读取出 N 个比特的集合,可以级联不同读取周期的 N 个比特的所得集合为比特流,该比特流形成全局帧的比特流。因此,即使在已经将全局帧的所有 B 个比特写入到存储器元件之前仍然可以出于数据传输的目的而执行使用 N 个存储器元件对 N 个交织的比特的写入和读取。这与仅使用一个单个存储器设备的先前描述的简单方法比较在大量程度上减少比特交织器的延时。

[0021] 进而另外,由于控制单元控制空间 - 时间交换机对不同数据流的读取、空间 - 时间交换机的交换和对用于将所得单个比特写入到存储器元件的比特子元件中的写入地址的选择,控制单元能够容易地改变用于不同数据流的不同数据速率,这又实现很大灵活性。

[0022] 概括上文,提出的比特交织器实现具有低延时和很大灵活性的高速比特交织。进而另外,由于仅需使用 N 个存储器元件这样的事实,所以与将一个大存储器用于在这一个大存储器内保持全局帧的所有 B 个比特的所提出的简单方法比较而减少功率消耗。

附图说明

[0023] 图 1a、图 1b 和图 1c 示出用于处于相同数据速率的不同数据流的比特交织原理。

[0024] 图 2a、图 2b 和图 2c 示出用于处于不同数据速率的不同数据流的比特交织原理。

[0025] 图 3a、图 3b 和图 3c 示出用于对于不同数据流实现相同数据速率的比特交织方法的在不同时刻具有用于不同存储器元件的不同存储器状态的所得输出矢量。

[0026] 图 4a、图 4b 和图 4c 示出用于对于不同数据流实现不同数据速率的比特交织方法的在不同时刻具有用于不同存储器元件的不同状态的所得的不同输出矢量。

[0027] 图 5 和图 6 示出通过从存储器元件读取比特集合而获得的所得数据流。

[0028] 图 7 示出根据一个优选实施例的提出的比特交织器。

具体实施方式

[0029] 图 7 示出提出的比特交织器 BI。

[0030] 比特交织器 BI 包含接收 M 个不同数据流 OD1…OD4 的 FIFO 读取器 FR。FIFO 读取器是可以用于接收和读取不同数据流 OD1、…、OD4 的存储器读取器的一个示例。可以代之以使用备选类型的存储器读取器。

[0031] 在这一示例中,选择数据流的数目 M 为 M = 4 而无任何限制。

[0032] FIFO 读取器 FR 向空间 - 时间交换机 STS 提供不同数据流 OD1…OD4。不同数据流 OD1、…、OD4 是向相应光网络单元指配的数据流。

[0033] 空间 - 时间交换机 STS 在读取周期内接收从相应数据流 OD1、…、OD4 提取的高达 M 个相应比特集合 BS1、…、BS4。比特集合 BS1、…、BS4 中的每个比特集合包含 K 个比特。在这一示例中,选择比特集合内的比特的数目 K 为 K = 8 而无任何限制。

[0034] 空间 - 时间交换机 STS 可以在一个读取周期内从数据流 OD1、…、OD4 中的每个数据流接收 K = 8 个比特的集合。在这一情况下,读取不同数据流 OD1、…、OD4 的速率对于所有数据流相同。为了实现用于不同数据流 OD1、…、OD4 的不同数据速率,空间 - 时间交换机 STS 可以在不同连续读取周期内从不同数据流 OD1、…、OD4 读取不同数目的比特集合。例如在第一读取周期内,空间 - 时间交换机 STS 可以从数据流 OD1、…、OD4 中的每个数据流读取相应比特集合 BS1、…、BS4,而在下一后继读取周期内,空间 - 时间交换机 STS

仅从数据流 OD1、OD2 读取相应比特集合 BS1、BS2。空间 - 时间交换机这样在不同读取周期内读取比特集合的结果是数据流 OD1、OD2 经历以如下数据速率的读取, 该数据速率是读取数据流 OD3、OD4 的数据速率的两倍。

[0035] 空间 - 时间交换机 STS 在一个写入周期内将上至 N 个比特交换到交换机 STS 的上至 N 个输出端口上。在 N 个输出端口中的每个输出端口, 提供相应输出矢量 v0、…、v(N-1)。随后将具体描述交换机 STS 将数据流的比特交换到输出矢量 v0、…、v(N-1) 上的方式。

[0036] 控制单元 CU 连接到交换机 STS 和读取器 FR。因此, 控制单元 CU 控制空间 - 时间交换机 STS 从读取器 FR 读取数据流。另外, 控制单元 CU 控制交换机 STS 将读取的比特交换到输出端口上并且因此交换到相应输出矢量 v0、…、v(N-1) 中的方式。

[0037] 然后向相应 OR 函数元件 00、…、0(N-1) 提供所得输出矢量 v0、…、v(N-1)。输出矢量 v0、…、v(N-1) 的比特长度为 J = 3。在这一示例中选择输出矢量的比特长度为 J = 3 而无任何限制。

[0038] OR 函数元件 00 基于输出矢量 v0 在一个写入周期内确定单个输出比特 b0。为此, OR 函数元件 00 通过布尔 OR 函数来组合在矢量 v0 内存在的所有比特。相应更多 OR 函数元件 01、…、0(N-1) 从相应矢量 v1、…、v(N-1) 确定相应单个输出比特 b1、…、b(N-1)。

[0039] 控制单元 CU 控制用于在一个读取周期内从相应 N 个存储器元件读取上至 N 个比特的读取地址。

[0040] 这样将输出矢量 v0、…、v(N-1) 减少成输出比特 b0、…、b(N-1) 有必要的原因是在一个写入周期内, 矢量 v0、…、v(N-1) 中的每个矢量仅包含交换机 STS 交换的一个比特。这一点的原因又是将空间 - 时间交换机 STS 用于将比特流的比特交换到在不同输出端口的输出矢量上是一种主要解决方案, 但是由于这样的交换机 STS 的内部调度性质, 不能确保在写入周期的每个时刻可以将数据流的比特之一交换到输出矢量 v0、…、v(N-1) 的恰好第一比特位置中。在该情况下, 比特集合 BS1、…、BS4 的并非所有比特在相同写入周期内被交换到交换机 STS 的相应输出端口上, 这样的比特序列 BS1、…、BS4 的一个或者多个比特必须被空间 - 时间交换机 STS 内部地延迟, 这又使这样的延迟的比特在输出矢量 v0、…、v(N-1) 之一以内被放置在与输出矢量的第一比特位置不同的比特位置。然而将具有输出矢量 v0、…、v(N-1) 的空间 - 时间交换机用于比特交织器 BI 是一种有利解决方案, 因为这样的交换机 STS 是用于从不同输入端口向不同输出端口交换比特的主要设备。交换的比特向在输出矢量内的所得比特位置的所得移位被 OR 函数元件 00、…、0(N-1) 补偿。

[0041] 然后向相应存储器元件 M0、…、M(N-1) 提供确定的单个输出比特 b0、…、b(N-1)。N 个存储器元件 M0、…、M(N-1) 中的每个存储器元件包括具有相应地址的至少 L 个比特子元件。在一个写入周期内, 存储器元件 M0、…、M(N-1) 在比特级向它的子元件之一写入相应单个输出比特 b0、…、b(N-1)。写入地址的如下选择由控制单元 CU 选择, 该选择确定存储器元件 M0、…、M(N-1) 向哪个比特子元件写入相应接收的输出比特 b0、…、b(N-1)。随后将具体描述写入地址的这一选择。

[0042] 提出的比特交织器 BI 能够在一个写入周期内在相应存储器元件 M0、…、M(N-1) 提供上至 N 个输出比特, 这些输出比特可以在一个读取周期内从这些存储器元件作为整体被读取出。因此, 在已经在一个写入周期内将上至 N 个比特写入到存储器元件 M0、…、M(N-1) 中之后, 这些 N 个比特然后可以恰在已经将第 N 个比特写入到存储器元件 M0、…、M(N-1)

之一中之后在读取周期内读取出。比特交织器 BI 无需在提取整个全局帧之前执行将全局帧的所有 B 个比特写入到大存储器中。

[0043] 比特交织器 BI 可以处理将各自在连续写入周期内的上至 N 个比特写入到存储器元件中、然后可以在用于形成对应全局帧的连续比特的后继读取周期中读取出 N 个比特的连续集合。因此,提出的比特交织器 BI 与在将全局比特交织帧的所有比特写入到整个存储器中、然后在已经将所有比特写入到单个大存储器中之后读取出整个全局帧时的解决方案相比实现减少的延时。

[0044] 进而另外,由于交换机 STS 读取数据流 OD1、…、OD4 的比特集合 BS1、…、BS4 的速率由控制单元 CU 控制,并且另外由于交换机 STS 的交换性质以及存储器元件 M0、…、M(N-1) 的读取和写入地址由控制单元 CU 控制,所以提出的比特交织器是一种灵活性很大的解决方案,该解决方案允许控制单元 CU 实现用于以高速比特交织的、具有低延时的不同数据流的不同数据速率。进而另外,由于仅使用 N 个存储器元件 M0、…、M(N-1),所以提出的比特交织器 BI 引起的功率消耗小于其中必须将 B 个比特的整个全局帧写入到保持这些 B 个比特的存储器中的解决方案。

[0045] 优选地,控制单元 CU 具有预定数据速率 R(i),其中数据流的索引 i 在这一示例中范围从 $i = 1, \dots, M = 4$ 。控制单元 CU 使用这些预定数据速率 R(i) 用于控制空间交换机 STS 从 FIFO 读取器 FR 读取比特集合 BS1、…、BS4。另外,控制单元 CU 具有预定偏移值 $\sigma(i)$ 而索引 i 定义不同数据流 OD1、…、OD4 的不同比特在全局帧内的偏移。随后将具体描述这些偏移的功能。

[0046] 进而另外,控制单元 CU 具有在全局帧内存在的比特的预定值 B 并且也具有存储器元件的数目 N 以及存储器的比特子元件的数目 L。因此,控制单元 CU 根据预定义的数据速率、预定义的偏移值、在帧内存在的比特的最大数目 B、存储器元件数目 N 和比特子元件数目 L 选择输出端口和写入地址。

[0047] 图 1a 将不同比特流 OD1、…、OD4 与作为随时间的序列的比特 A1、…、A8、B1、…、B8、C1、…、C8、D1、…、D8 一起示出。

[0048] 图 1b 示出全局帧 GF 的比特交织模式,其中交织来自图 1a 的数据流的不同比特。图 1b 示出用于该情况的数据流的不同比特的交织,数据速率 R(1)、R(2)、R(3)、R(4) 对于所有数据流都等于 $1/8$ 。换而言之,所有数据速率具有相同速率 R(1)、…、R(4)。在全局帧 GF 内存在的比特的最大数目 B 可以例如是 1,228,800 个比特。这样的全局帧可以具有 122 微秒的持续时间。

[0049] 数据流 OD1 的第一比特——该第一比特是比特 A1——被放置在全局帧 GF 的比特位置 0。由于为比特流 OD1 而选择的速率 R(1) 是 $1/8$ 这一事实,数据流 OD1 的下一比特 A2 被放在从第一比特 A1 隔开 8 个其他比特,从而比特 A2 被放置在比特位置 8。数据流 OD1 的其他比特 A3、A4、A5 在全局帧 GF 内的接下来比特位置 16、24、32 跟随。

[0050] 对于在全局帧内放置比特的所示 BIPON 比特交织方案,一般可以通过使用速率 $R(i)$ 的倒数值 $r(i) = 1/R(i)$ 、偏移值 $\sigma(i)$ 和整数值 $x(i)$ 来确定具有索引 i 的数据流的第 x 个比特的比特位置 $bp(x(i), i)$ 。整数值 $x(i)$ 对应于第 x 个比特的索引号、因此是范围从 $x(i) = 0$ 到整数值 $x(i) = \text{int}(B/r(i))$ 的整数。因此,可以对于十进制数确定具有索引 i 的数据流的第 x 个比特的比特位置 $bp(x(i), i)$ 为:

[0051] $bp(x(i), i) = o(i) + r(i) * x(i)$ 。

[0052] 因此,控制单元 CU 如以上具体描述的那样对于第 i 个数据流的第 x 个比特确定在全局帧内的对应比特位置。

[0053] 在这一示例中,对于数据流 OD1 选择偏移值 $o(1)$ 为 $o(1) = 0$ 。

[0054] 在图 1b 中,也示出数据流 OD4 的比特 D1、D2、…在全局帧 GF 内的关系。由于该事实,对于这一比特交织方案也选择速率 $R(4)$ 为 $R(4) = 1/8$,数据流 OD4 的比特 D1、D2、…各自在全局帧 GF 内被间隔开 8 个比特位置。在这一示例中选择偏移值 $o(4)$ 为值 $o_4 = 3$ 。在图 1 中未明示用于数据流 OD2、OD3 的比特的偏移值,但是可以认为选择偏移值 $o(2)$ 为值 $o(2) = 1$ 而选择偏移值 $o(3)$ 为值 $o(3) = 2$ 。

[0055] 查看图 1b,变得清楚的是接收这一全局帧 GF 的光网络单元可以与除以 8 的全局帧 GF 的数据速率相等的数据速率执行对数据流之一的比特的接收。

[0056] 图 1c 示出备选比特模式 ABP,其中图 1b 中所示全局帧的不同比特被拆分成 N 个比特的集合。随时间可以从图 7 中所示存储器设备 $M_0, \dots, M(N-1)$ 读取出并且随后成功地发送 N 个比特的不同集合,以用于实现发送与图 1b 中所示全局帧 GF 对应的比特序列。

[0057] 必须在第一读取周期中从图 7 中所示存储器 $M_0, \dots, M(N-1)$ 读取的第一集合是集合 S_1 。这一集合 S_1 的比特对应于在图 1b 的全局帧 GF 的比特位置 0 到 15 内存储的那些比特。

[0058] 在下一读取周期内,应当从图 7 中所示存储器元件 $M_0, \dots, M(N-1)$ 读取出图 1c 中所示下一比特集合 S_2 。这些是向图 1b 中所示全局帧 GF 的比特位置 16 到 31 分配的比特。

[0059] 在第三读取周期内,应当从图 7 中所示存储器元件 $M_0, \dots, M(N-1)$ 读取出图 1c 中所示比特集合 S_3 。这些是在图 1b 中所示全局帧 GF 的比特位置 32 开始的其他比特。

[0060] 现在将具体说明交换机 STS 将交换并且被写入到图 7 中所示存储器 $M_0, \dots, M(N-1)$ 中并且另外也从这些存储器读取不同数据流的不同比特的方式。

[0061] 应当假设交换机 STS 在一个输入周期内从光纤读取器 FR 读取各自为 $K = 8$ 个比特的比特集合 BS_1, \dots, BS_4 。在写入周期内,空间时间交换机 STS 将比特集合 BS_1, \dots, BS_4 的第一和第二比特交换到相应输出端口上并且因此交换到如图 3a 中所示的相应矢量 v_0, \dots, v_{15} 中。由于交换机 STS 的内部调度性质,交换的比特存在于输出矢量 v_0, \dots, v_{15} 内的具有索引 0 的第一比特位置内。作为示例,选择输出矢量 v_0, \dots, v_{15} 的比特长度为 $J = 3$ 作为非限制示例。

[0062] 交换机 STS 将具体比特交换到的输出端口的编号和因此交换到的对应输出矢量的编号由控制单元 CU 控制。控制单元 CU 确定比特被交换到的输出端口的编号以及因此交换到的输出矢量的编号和也交换到的存储器的所得编号 $M_0, \dots, M(N-1)$ 对于十进制数为端口编号 $pn(x(i), i) = bp(x(i), i) \bmod N$ 。对于每个比特,比特位置 bp 可以如以上先前描述的那样由控制单元 CU 确定。

[0063] 图 3a 还示出不同存储器元件 M_0, \dots, M_{15} 在时刻 $t = 0$ 的不同状态。不同 OR 函数元件将不同输出矢量 v_0, \dots, v_{15} 减少成单个输出比特 $b_0, \dots, b_{(N-1)}$,然后向相应存储器 $M_0, \dots, M(N-1)$ 的相应比特子元件中写入该单个输出比特 $b_0, \dots, b_{(N-1)}$,其中这些比特子元件具有相同写入地址。在这一示例中,写入地址等于 0。控制单元 CU 在一个写入周期内对于第 i 个数据流的第 x 个比特根据第 i 个数据流的第 x 个比特的比特位置和数目 L

确定写入地址 wa。具体而言,可以对于十进制数确定第 i 个数据流的第 x 个比特的写入地址 wa 为:

[0064] $wa(x(i), i) = \text{int}(bp(x(i), i)/L)$ 。

[0065] 查看在时刻 $t = 0$ 的存储器元件 M0、…、M15,可以用相同读取地址在一个时刻读取出所有存储器 M0、…、M15 来从存储器读取图 1c 中示出为比特集合 S1 的 N 个比特的适当集合。在此示例中,读取地址等于 0。

[0066] 无需确切地在比特最终已经被写入到存储器元件 M0、…、M15 中的时刻执行、而是可以在更晚时间点执行这一读取。用适当读取地址从存储器元件 M0、…、M15 读取比特由图 7 中所示控制单元 CU 控制和确定。

[0067] 在下一写入周期内,交换机将不同数据流的比特交换到输出端口上和如图 3b 中所示的对应输出矢量 v0、…、v15 中。在矢量 v0、…、v15 内的比特位置 0 放置比特 A3、…、D4 由交换机的内部调度性质引起。

[0068] 如先前描述的那样,图 7 中所示提到的 OR 函数元件 00、…、0(N-1) 如图 3b 中所示生成然后在这一时刻 $t = 1$ 向存储器元件 M0、…、M15 的相应比特子元件中写入的单个输出比特。

[0069] 在这一下一写入周期内,控制单元 CU 对于第 i 个数据流的第 x 个比特根据比特位置 $bp(x(i), i)$ 确定所得输出比特被写入到的输出端口的编号和因此写入到的存储器元件的编号为端口编号 pn。具体而言,控制单元 CU 对于十进制数、针对第 i 个数据流的第 x 个比特确定端口编号 pn 为:

[0070] $pn(x(i), i) = bp(x(i), i) \bmod N$ 。

[0071] 另外,使用不同比特 A3、…、D4 的相应比特位置,图 7 的控制单元 CU 对于十进制数确定写入地址 w 为:

[0072] $wa(x(i), i) = \text{int}(bp(x(i), i)/L)$ 。

[0073] 通过查看图 3b,变得清楚的是可以通过在相同读取地址读取出存在于存储器元件 M0、…、M15 内的 16 个比特来从这些存储器元件读取出 N = 16 个比特的下一集合,这些读取地址在这一示例中是值为 1 的读取地址。

[0074] 因此,在必须从不同存储器元件读取 N 个比特的连续集合时,为了形成与 BIPON 方案的全局帧对应的比特序列,可以简单地在相同读取周期内对于所有这些 N 个存储器元件使用相同读取地址从 N 个存储器元件读取出这样的 N 个比特、然后简单地增加用于下一读取周期的读取地址。

[0075] 对于其他比特 A5、…、D6,图 3c 对于下一时刻 $t = 2$ 示出用于交换这些比特并且将这些比特写入到存储器 M0、…、M15 中的提出的选择造成在存储器元件内的具体比特子元件放置比特的方式。

[0076] 图 5 中所示所得比特序列 RBS 从在每个相应读取周期内用相同读取地址从 N 个存储器元件读取 N 个比特的集合而产生。通过比较图 5 的所得比特序列 RBS 与图 1b 中的全局帧 GF 的初始地希望的比特序列,变得清楚的是图 7 中的提出的比特交织器 BI 能够执行图 1b 中所示全局帧的希望的比特交织。

[0077] 图 2a 同样示出不同数据流 OD1、…、OD4。

[0078] 图 2b 示出在如下情况下的全局帧 GF2 的希望的比特序列,该情况是应当以速率

1/8 交织数据流 OD1 并且也应当以速率 1/8 交织数据流 OD4, 而应当以速率 1/16 交织数据流 OD2 并且应当以速率 1/32 交织数据流 OD3。

[0079] 如先前关于图 1b 描述的那样选择不同数据流的偏移值。

[0080] 图 2c 随时间示出具有 $N = 16$ 个比特的集合的对应备选比特模式 ABP2。所得比特集合 S1、S12、S13 如图 7 中所示应当在存储器元件 M0、…、M(N-1) 内存在于相同地址的比特子元件。

[0081] 图 4a 示出不同比特 A1、…、D2 在输出矢量 v0、…、v15 以及存储器元件 M0、…、M15 内的所得放置。

[0082] 在下一时刻 $t = 1$, 必须如图 4b 中所示在输出矢量 v0、…、v15 内放置图 2c 中所示集合 S12 的比特。

[0083] 由于图 7 中所示空间时间交换机 STS 必须交换比特 B2 的事实, 这一交换机 STS 的内部调度性质引起这一比特 B2 放置在输出矢量 v1 中的值为 1 的比特位置。如先前具体描述的那样, OR 函数元件如图 7 中所示实现将相应输出矢量 v0、…、v15 减少成相应输出比特 b0、…、b(N-1)。

[0084] 然后在与时刻 $t = 1$ 对应的一个写入周期内将所得单个输出比特写入到写入地址为 $wa(x(i), i) = \text{int}(bp(x(i), i)/L)$ 的比特子元件中。

[0085] 对于下一另一时刻 $t = 2$, 图 4c 示出不同比特放置于输出矢量 v0、…、v15 内以及存储器元件 M0、…、M15 的等于值为 2 的相同地址的比特子元件内。

[0086] 通过查看在右手侧的图 4c, 变得清楚的是现在有可能在连续读取周期从不同存储器元件 M0、…、M15 读取出 N 个比特的相应集合, 其中在一个读取周期内, 相同读取地址用于选择每个存储器元件 M0、…、M15 的比特子元件。

[0087] 通过如以上先前描述的那样在连续读取周期内从存储器元件读取出 N 个比特的集合, 可以实现如图 6 中所示所得比特模式 RBS2。通过比较图 6 的比特序列 RBS2 与图 4b 的全局帧 GF2 的希望的比特序列, 变得清楚的是图 7 中所示提出的比特交织器 BI 实现 BIPON 需要的希望的比特交织方法。

[0088] 选择比特子元件数目为 $L = 3$ 仅为示例选择。优选地, 选择在存储器元件内存在的比特子元件数目为 $L = 1024$ 。

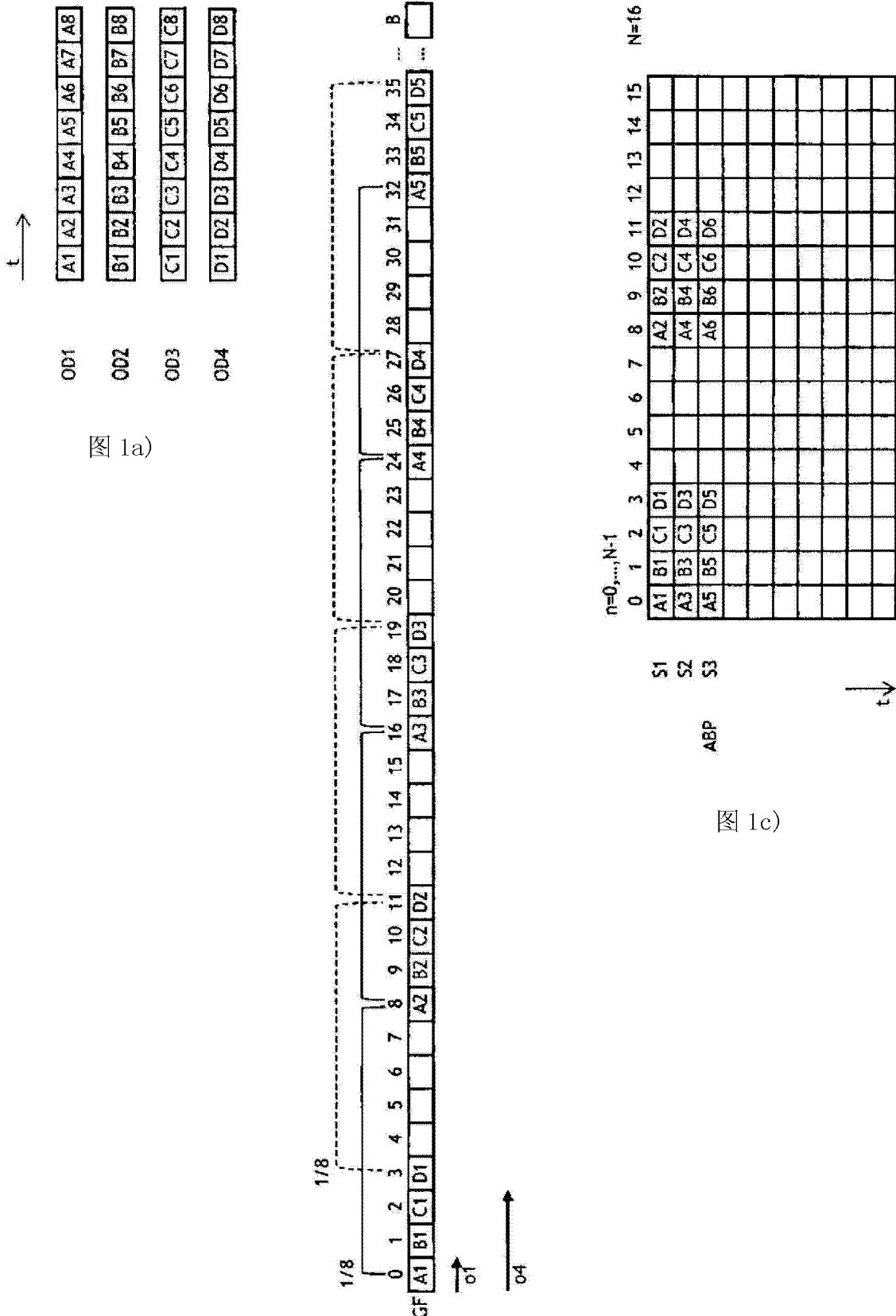
[0089] 为了实现每秒 10 吉比特的总数据速率, 可以选择数目 N 为 64 而以速率或者频率 $f = 156, 25\text{MHz}$ 读取出 N 个比特的这样的集合。作为备选示例, 可以频率 $f = 312, 5\text{MHz}$ 写入和读取 $N = 32$ 比特的集合用于实现每秒 10 吉比特的相同比特率。进而备选地, 可以选择数目 N 为 $N = 128$ 而读取频率为 $f = 78, 125\text{MHz}$ 。

[0090] 图 7 中所示提出的比特交织器 BI 的明显优点是系统的全局延时低, 因为可以在完成形成全局帧的所有比特的所有交织过程之前开始从存储器元件 M0、…、M(N-1) 读取交织的比特的过程。

[0091] 优选地, 在存储器读取器是 FIFO 读取器的情况下, 在单个现场可编程门阵列 (FPGA) 内实施图 7 中所示比特交织器 BI 的不同子设备。

[0092] 可以通过使用专用硬件以及能够与适当软件关联地执行软件的硬件来提供图 7 中所示各种单元的功能, 这些单元包括标注为‘控制单元’的任何功能块。在由处理器提供时, 功能可以由单个专用处理器、由单个共享处理器或者由多个个别处理器提供, 这些个别

处理器中的一些处理器可以被共享。另外，不应解释对术语‘控制单元’的显式使用仅指代能够执行软件的硬件而可以隐含地包含而不限于数字信号处理器 (DSP) 硬件、网络处理器、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA)、用于存储软件的只读存储器 (ROM)、随机存取存储器 (RAM) 和非易失性存储装置。也可以包括其它常规和 / 或定子硬件。相似地，图中所示任何交换机仅为概念。可以通过程序逻辑的操作、通过专用逻辑、通过程序控制和专用逻辑的交互或者甚至手动地执行它们的功能，具体技术可由实施者如从上下文更具体地理解的那样选择。



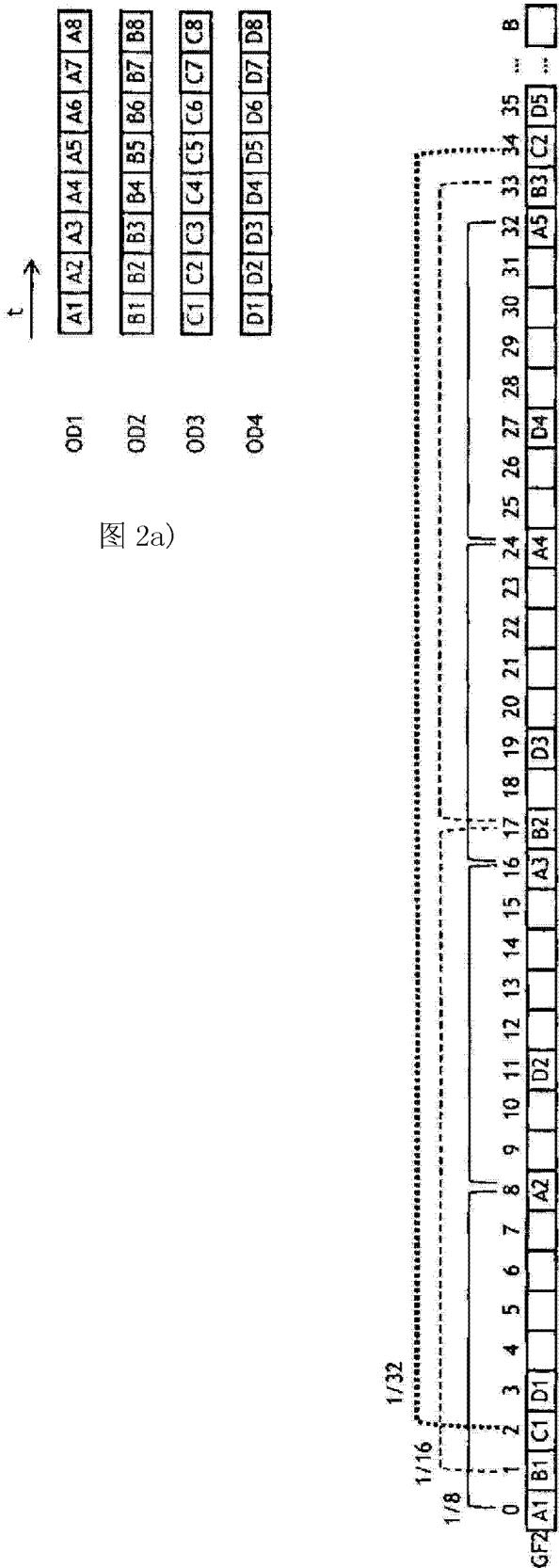


图 2b)

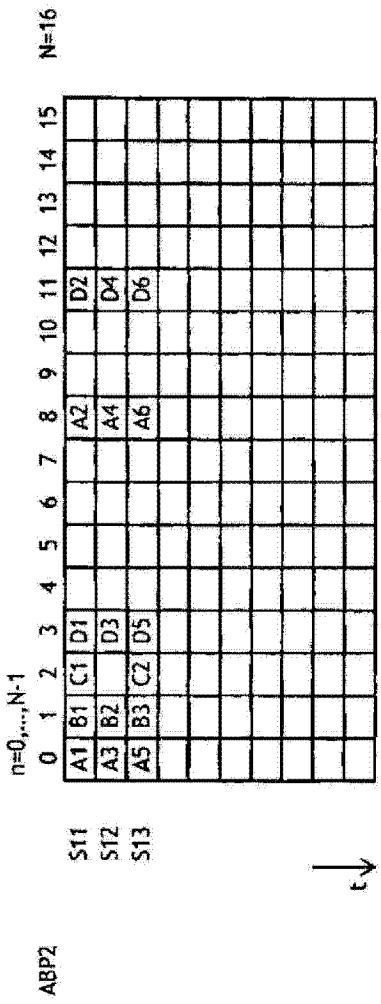


图 2c)

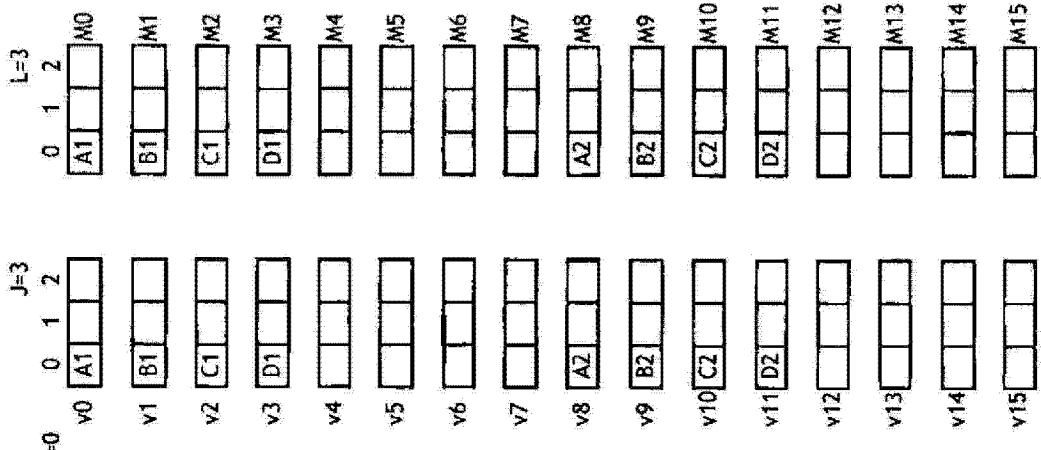


图 3a)

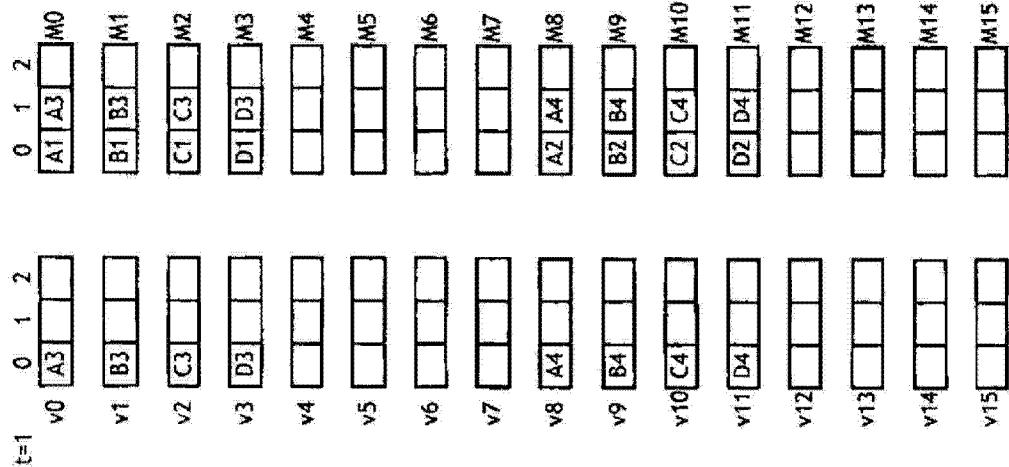


图 3b)

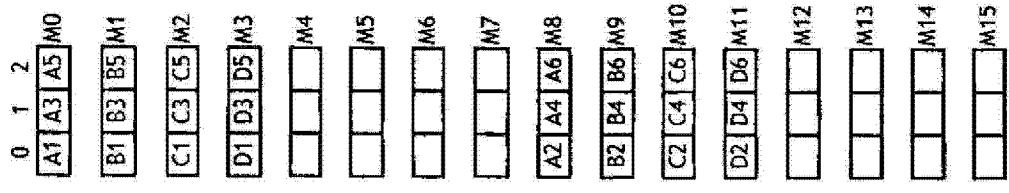


图 3c)

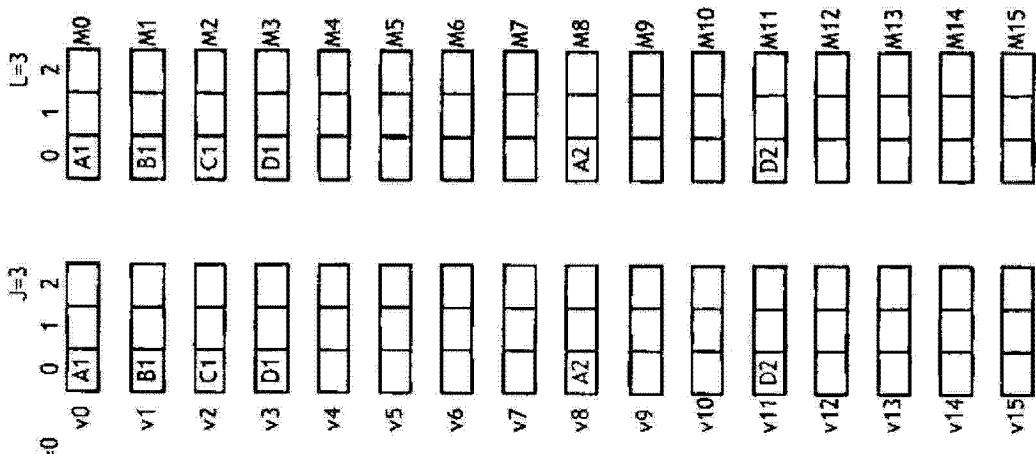


图 4a)

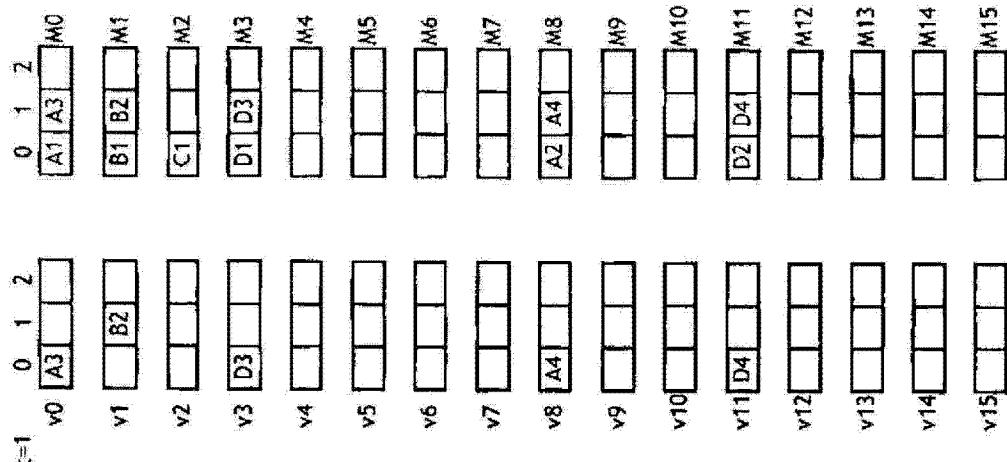


图 4b)

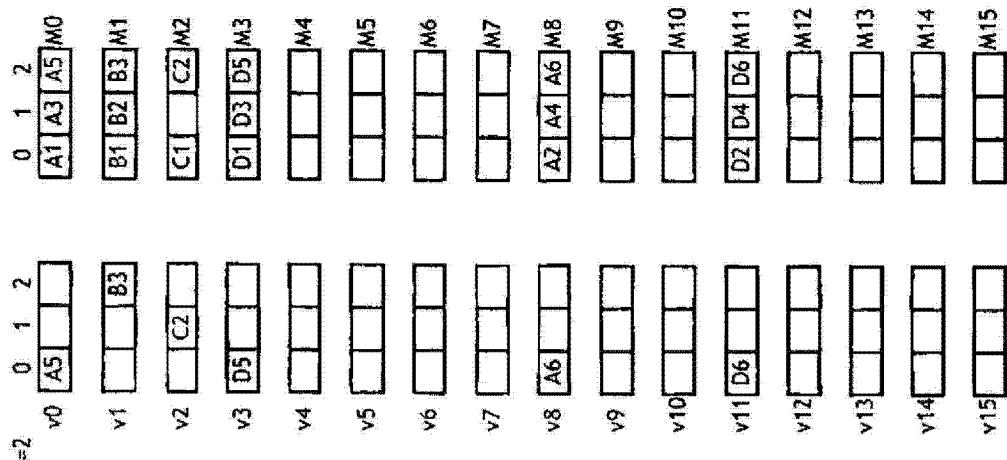


图 4c)

读取地址=0															
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15															
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15															
A1 B1 C1 D1 A2 B2 C2 D2 A3 B3 C3 D3 A4 B4 C4 D4 A5 B5 C5 D5 ...															
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 ... B															

图 5

读取地址=1															
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15															
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15															
A1 B1 C1 D1 A2 B2 C2 D2 A3 B3 C3 D3 A4 B4 C4 D4 A5 B5 C5 D5 ...															
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 ... B															

图 6

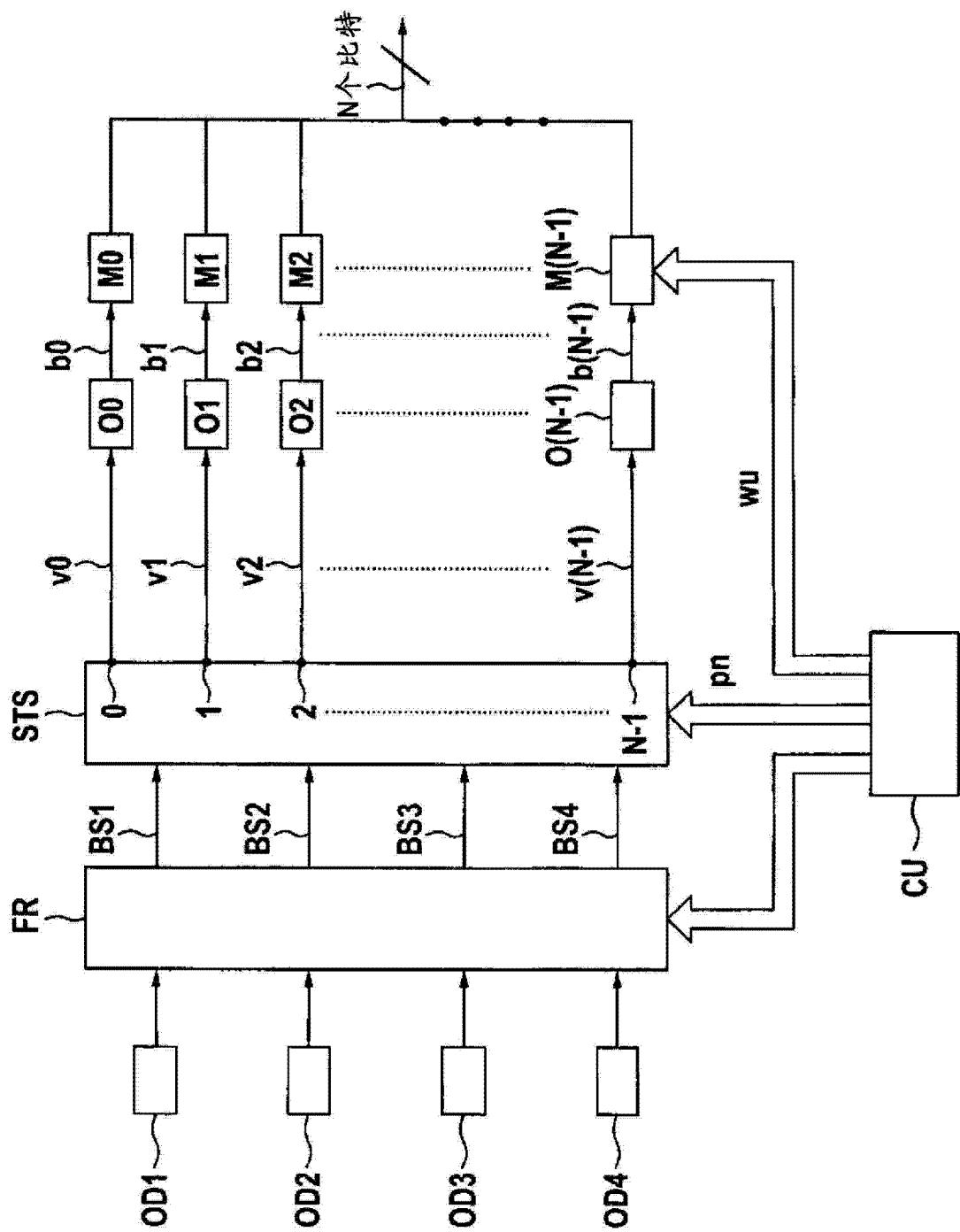


图 7