



(12) 发明专利

(10) 授权公告号 CN 102331922 B

(45) 授权公告日 2015. 07. 15

(21) 申请号 201110145879. 5

(22) 申请日 2011. 06. 01

(30) 优先权数据

2010-130447 2010. 06. 07 JP

(73) 专利权人 佳能株式会社

地址 日本东京

(72) 发明人 伊藤忠幸

(74) 专利代理机构 中国国际贸易促进委员会专
利商标事务所 11038

代理人 罗银燕

(51) Int. Cl.

G06F 9/38(2006. 01)

G06F 12/08(2006. 01)

(56) 对比文件

US 5572690 A, 1996. 11. 05, 说明书第 6-7
栏、图 1B.

US 5748539 A, 1998. 05. 05, 说明书第 3-7
栏、图 1、图 5B.

US 7546405 B2, 2009. 06. 09, 说明书第 4 栏.
CN 1431599 A, 2003. 07. 23, 全文.

US 2010/0106910 A1, 2010. 04. 29, 说明书第
[0012] 段.

US 5710905 A, 1998. 01. 20, 说明书第 4 栏、
图 1-2.

审查员 狄晓斐

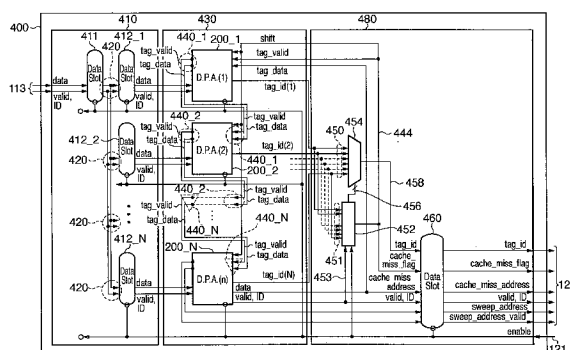
权利要求书4页 说明书17页 附图11页

(54) 发明名称

运算装置、高速缓存装置及其控制方法

(57) 摘要

本发明涉及运算装置、高速缓存装置及其控制方法。提供一种装置，该装置包括：第一保持单元，该第一保持单元中的每一个包含串行连接的第一节点，并且在第一方向上移动各第一节点中的第一数据；第二保持单元，该第二保持单元中的每一个包含串行连接的第二节点，并且在第二方向上移动各第二节点中的第二数据。各第一节点与第二节点中的至少一个对应。该装置还包括：运算单元，该运算单元对于作为第一节点的关注节点，使用关注节点中的第一数据和关注节点所对应的第二节点中的至少一个中的第二数据执行运算；以及输入单元，该输入单元将第一数据并行输入到第一保持单元之中的至少两个，并且将第二数据串行输入到第二保持单元之中的至少两个。



1. 一种运算装置,包括:

多个第一保持单元,所述多个第一保持单元中的每一个包含串行连接的多个第一节点,并且将保持在各第一节点中的第一数据移动到位于第一方向上的相邻节点;

多个第二保持单元,所述多个第二保持单元中的每一个包含串行连接的多个第二节点,并且将保持在各第二节点中的第二数据移动到位于第二方向上的相邻节点,其中,各第一节点与所述多个第二节点中的至少一个对应,并且所述多个第二保持单元被串行连接,以使得保持在位于一个第二保持单元的最下游的第二节点中的第二数据被移动到位于另一第二保持单元的最上游的第二节点;

运算单元,所述运算单元对于作为第一保持单元的所述多个第一节点之一的关注节点,使用保持在关注节点中的第一数据和保持在关注节点所对应的所述多个第二节点中的至少一个中的第二数据来执行运算;以及

输入单元,所述输入单元将复制的第一数据和原始的第一数据并行输入到所述多个第一保持单元之中的至少两个,并且将第二数据输入到所述多个第二保持单元之一,其中所输入的第二数据被移动通过所述多个第二保持单元,其中至少两个同样的第一数据与保持于所述多个第二保持单元中的不同的第二数据一起被用于由运算单元执行的各个运算。

2. 根据权利要求 1 的运算装置,还包括:

获取单元,所述获取单元从外部装置获取第一数据和第二数据;

第一输入单元,所述第一输入单元复制获取的第一数据,并且将复制的第一数据和原始的第一数据输入到所述多个第一保持单元中的每一个;以及

第二输入单元,所述第二输入单元将获取的第二数据输入到所述多个第二保持单元之一。

3. 根据权利要求 1 的运算装置,还包括:

多个运算电路,所述多个运算电路中的每一个包含一个第一保持单元和一个第二保持单元,

其中,所述多个运算电路被分组成多个组,以及

输入单元对于各组将第二数据输入到组中的一个运算电路的第二保持单元,并且在各组中,第二保持单元被串行连接。

4. 根据权利要求 3 的运算装置,还包括:

动态改变所述组的单元。

5. 一种确定装置,包括根据权利要求 1 的运算装置,其中:

各第一保持单元是第一流水线,所述第一流水线包含多个第一节点,并且将保持在一个第一节点中的第一数据移动到相对于所述一个第一节点位于第一方向上的另一第一节点;

各第二保持单元是第二流水线,所述第二流水线包含多个第二节点,并且将保持在一个第二节点中的第二数据移动到相对于所述一个第二节点位于与第一方向相反的第二方向上的另一第二节点;以及

运算单元是比较单元,所述比较单元对于作为第一流水线的所述多个第一节点之一的关注节点,比较保持在关注节点中的第一数据是否与保持在关注节点所对应的所述多个第二节点中的至少一个中的第二数据匹配。

6. 一种高速缓存装置,所述高速缓存装置包括多个数据处理装置,所述多个数据处理装置中的每一个包括:

第一流水线,所述第一流水线包含多个第一节点,并且将保持在一个第一节点中的地址移动到相对于所述一个第一节点位于第一方向上的另一第一节点;

第二流水线,所述第二流水线包含多个第二节点,并且将保持在一个第二节点中的高速缓存标记移动到相对于所述一个第二节点位于与第一方向相反的第二方向上的另一第二节点,其中,各第一节点与所述多个第二节点中的至少两个对应;以及

比较单元,所述比较单元对于作为第一流水线的所述多个第一节点之一的关注节点,比较保持在关注节点中的地址与保持在关注节点所对应的第二节点中的至少两个中的高速缓存标记,以确定地址是否与高速缓存标记匹配;所述高速缓存装置还包括:

高速缓存存储器,所述高速缓存存储器对存储在外部存储器件中的数据中的一些进行存储;

获取单元,所述获取单元从处理装置获取数据的请求;

第一输入单元,所述第一输入单元复制获取的数据的地址,并且将复制的地址和原始的地址输入到各自的数据处理装置的第一流水线;

第二输入单元,所述第二输入单元将存储于高速缓存存储器中的数据的高速缓存标记输入到所述多个数据处理装置之一的第二流水线;以及

确定单元,所述确定单元当确定比较单元的比较结果指示匹配时,确定获取的数据的高速缓存命中,

其中,所述多个数据处理装置的第二流水线被串行连接,以使得保持在位于一个第二流水线的最下游的第二节点中的高速缓存标记被移动到位于另一第二流水线的最上游的第二节点,并且该高速缓存标记被移动通过所述多个数据处理装置的第二流水线,以及

其中,至少两个同样的地址和保持于所述多个数据处理装置的第二流水线中的不同的高速缓存标记被比较。

7. 根据权利要求 6 的高速缓存装置,其中,所述多个数据处理装置被分成多个组,

第二输入单元对于各组将高速缓存标记输入到组中的一个数据处理装置的第二流水线,以及

在各组中,第二流水线被串行连接。

8. 根据权利要求 7 的高速缓存装置,还包括动态改变所述组的单元。

9. 根据权利要求 7 的高速缓存装置,其中,高速缓存存储器被分成与所述多个数据处理装置分别对应的区域,以及

与不包含于所述多个组中的任一个中的数据处理装置对应的高速缓存存储器的区域被暂停。

10. 根据权利要求 6 的高速缓存装置,其中,当驱动信号有效时,第一流水线将保持在所述多个第一节点中的每一个中的地址移动到第一方向上位于下游侧的相邻的第一节点,以及

当驱动信号有效并且确定单元确定高速缓存遗漏时,第二流水线将保持在所述多个第二节点中的每一个中的高速缓存标记移动到第二方向上位于下游侧的相邻的第二节点。

11. 根据权利要求 6 的高速缓存装置,其中,当确定单元确定高速缓存遗漏时,第二输

入单元将从第一流水线输出的地址输入到第二流水线作为高速缓存标记。

12. 根据权利要求 6 的高速缓存装置, 其中, 来自处理装置的数据的请求是对于外部存储器件的数据的写入请求, 以及

高速缓存装置还包括使用从第二流水线输出的高速缓存标记在高速缓存存储器中写入与写入请求对应的数据的单元。

13. 一种运算装置的控制方法, 所述运算装置包括:

多个第一保持单元, 所述多个第一保持单元中的每一个包含串行连接的多个第一节点, 并且将保持在各第一节点中的第一数据移动到位于第一方向上的相邻节点,

多个第二保持单元, 所述多个第二保持单元中的每一个包含串行连接的多个第二节点, 并且将保持在各第二节点中的第二数据移动到位于第二方向上的相邻节点, 其中, 各第一节点与所述多个第二节点中的至少一个对应, 并且所述多个第二保持单元被串行连接, 以使得保持在位于一个第二保持单元的最下游的第二节点中的第二数据被移动到位于另一第二保持单元的最上游的第二节点, 以及

运算单元, 所述运算单元对于作为第一保持单元的所述多个第一节点之一的关注节点, 使用保持在关注节点中的第一数据和保持在关注节点所对应的所述多个第二节点中的至少一个中的第二数据来执行运算, 所述控制方法包括:

将复制的第一数据和原始的第一数据并行输入到所述多个第一保持单元之中的至少两个, 并且将第二数据输入到所述多个第二保持单元之一, 其中所输入的第二数据被移动通过所述多个第二保持单元,

其中至少两个同样的第一数据与保持于所述多个第二保持单元中的不同的第二数据一起被用于由运算单元执行的各个运算。

14. 一种高速缓存装置的控制方法, 所述高速缓存装置包括多个数据处理装置, 所述多个数据处理装置中的每一个包括:

第一流水线, 所述第一流水线包含多个第一节点, 并且将保持在一个第一节点中的地址移动到相对于所述一个第一节点位于第一方向上的另一第一节点,

第二流水线, 所述第二流水线包含多个第二节点, 并且将保持在一个第二节点中的高速缓存标记移动到相对于所述一个第二节点位于与第一方向相反的第二方向上的另一第二节点, 其中, 各第一节点与所述多个第二节点中的至少两个对应, 以及

比较单元, 所述比较单元对于作为第一流水线的所述多个第一节点之一的关注节点, 比较保持在关注节点中的地址与保持在关注节点所对应的第二节点中的至少两个中的高速缓存标记, 以确定地址是否与高速缓存标记匹配, 所述控制方法包括:

在高速缓存存储器中存储被存储在外部存储器件中的数据中的一些;

控制获取单元以从处理装置获取数据的请求;

控制第一输入单元以复制获取的数据的地址并且将复制的地址和原始的地址输入到各自的数据处理装置的第一流水线;

控制第二输入单元以将存储于高速缓存存储器中的数据的高速缓存标记输入到所述多个数据处理装置之一的第二流水线; 以及

控制确定单元以在确定比较单元的比较结果指示匹配时确定获取的数据的高速缓存命中,

其中,所述多个数据处理装置的第二流水线被串行连接,以使得保持在位于一个第二流水线的最下游的第二节点中的高速缓存标记被移动到位于另一第二流水线的最上游的第二节点,并且该高速缓存标记被移动通过所述多个数据处理装置的第二流水线,

其中,至少两个同样的地址和保持于所述多个数据处理装置的第二流水线中的不同的高速缓存标记被比较。

运算装置、高速缓存装置及其控制方法

技术领域

[0001] 本发明涉及运算装置、高速缓存 (cache) 装置及其控制方法。

背景技术

[0002] 存在要求确定包含于第一数据组中的各单个数据是否与第二数据组中的至少一些数据匹配的许多应用。日本专利 No. 3588487 提出了逆流流水线 (counterflow pipeline) 技术以获得这种确定。以下将参照图 11 描述逆流流水线技术。逆流流水线技术使用作为两个流水线的第一流水线 1110 和第二流水线 1120。第一流水线 1110 具有多个第一节点 1111, 并且这些第一节点 1111 在一定的定时存储第一数据组的数据 A[0] 至 A[3]。第一流水线 1110 沿右方向移动 (shift) 数据。作为数据移动的结果, 例如, 存储数据 A[2] 的第一节点 1111 存储数据 A[3]。第二流水线 1120 具有多个第二节点 1121, 并且这些第二节点 1121 在一定的定时存储第二数据组的数据 B[0] 至 B[3]。第二流水线 1120 沿左方向移动数据。确定单元 1130 与第一节点 1111 和第二节点 1121 连接, 并且确定存储于这些节点中的数据是否匹配。对于逆流流水线技术, 由于两个流水线中的数据沿相反的方向移动, 因此, 通过循环 (round-robin) 方法比较第一组和第二组的数据。

[0003] 为了获得较快的逆流流水线运算, 两个流水线独立地移动。但是, 当第一流水线 1110 和第二流水线 1120 同时移动时, 一些数据不能被比较。例如, 在图 11 所示的定时, 比较数据 A[2] 和 B[1]。当第一和第二流水线同时移动时, 在下一定时比较数据 A[2] 和数据 B[3]。因此, 数据 A[2] 从不与数据 B[2] 比较。为了抑制这样的数据比较差错, 在日本专利 No. 3588487 中, 监视第一流水线 1110 和第二流水线 1120 的节点的状态。在将出现数据比较差错的状态中, 两个流水线被停止, 并且在完成比较之后被允许移动数据。并且, 在日本专利 No. 3588487 中描述的逆流流水线技术中, 数据仅经受一对一的比较处理, 但一个数据不能同时与多个数据比较, 或者, 不能动态地改变要与一个数据同时比较的数据的数量。

发明内容

[0004] 如上所述, 日本专利 No. 3588487 中描述的逆流流水线技术难以确实地 (surely) 且灵活地比较数据。由此, 本发明的一个方面提供一种用于使用逆流流水线确实地且灵活地比较数据的技术。

[0005] 本发明的一个方面提供一种运算装置, 该运算装置包括: 多个第一保持单元, 所述多个第一保持单元中的每一个包含串行连接的多个第一节点, 并且将保持在各第一节点中的第一数据移动到位于第一方向上的相邻节点; 多个第二保持单元, 所述多个第二保持单元中的每一个包含串行连接的多个第二节点, 并且将保持在各第二节点中的第二数据移动到位于第二方向上的相邻节点, 其中, 各第一节点与所述多个第二节点中的至少一个对应; 运算单元, 所述运算单元对于作为第一保持单元的所述多个第一节点之一的关注节点, 使用保持在关注节点中的第一数据和保持在关注节点所对应的所述多个第二节点中的至少一个中的第二数据来执行运算; 以及输入单元, 所述输入单元将第一数据并行输入到所述

多个第一保持单元之中的至少两个,并且将第二数据串行输入到所述多个第二保持单元之中的至少两个。

[0006] 本发明的另一方面提供一种高速缓存装置,所述高速缓存装置包括多个数据处理装置,所述多个数据处理装置中的每一个包括:第一流水线,所述第一流水线包含多个第一节点,并且将保持在一个第一节点中的地址移动到相对于所述一个第一节点位于第一方向上的另一第一节点;第二流水线,所述第二流水线包含多个第二节点,并且将保持在一个第二节点中的高速缓存标记移动到相对于所述一个第二节点位于与第一方向相反的第二方向上的另一第二节点,其中,各第一节点与所述多个第二节点中的至少一个对应;以及比较单元,所述比较单元对于作为第一流水线的所述多个第一节点之一的关注节点,比较保持在关注节点中的地址与保持在关注节点所对应的第二节点中的至少一个中的高速缓存标记,以确定地址是否与高速缓存标记匹配;所述高速缓存装置还包括:高速缓存存储器,所述高速缓存存储器对存储在外部存储器件中的数据中的一些进行存储;获取单元,所述获取单元从处理装置获取数据的请求;第一输入单元,所述第一输入单元复制获取的数据的地址,并且将复制的地址输入到所述多个数据处理装置的第一流水线;第二输入单元,所述第二输入单元将存储于高速缓存存储器中的数据的高速缓存标记输入到所述多个数据处理装置之一的第二流水线;以及确定单元,所述确定单元当确定比较单元的比较结果指示匹配时,确定获取的数据的高速缓存命中(hit),其中,所述多个数据处理装置的第二流水线被串行连接。

[0007] 本发明的又一方面提供一种运算装置的控制方法,所述运算装置包括:多个第一保持单元,所述多个第一保持单元中的每一个包含串行连接的多个第一节点,并且将保持在各第一节点中的第一数据移动到位于第一方向上的相邻节点;多个第二保持单元,所述多个第二保持单元中的每一个包含串行连接的多个第二节点,并且将保持在各第二节点中的第二数据移动到位于第二方向上的相邻节点,其中,各第一节点与所述多个第二节点中的至少一个对应;以及运算单元,所述运算单元对于作为第一保持单元的所述多个第一节点之一的关注节点,使用保持在关注节点中的第一数据和保持在关注节点所对应的所述多个第二节点中的至少一个中的第二数据来执行运算,所述方法包括:将第一数据并行输入到所述多个第一保持单元之中的至少两个,并且将第二数据串行输入到所述多个第二保持单元之中的至少两个。

[0008] 本发明的再一方面提供一种高速缓存装置的控制方法,所述高速缓存装置包括多个数据处理装置,所述多个数据处理装置中的每一个包括:第一流水线,所述第一流水线包含多个第一节点,并且将保持在一个第一节点中的地址移动到相对于所述一个第一节点位于第一方向上的另一第一节点;第二流水线,所述第二流水线包含多个第二节点,并且将保持在一个第二节点中的高速缓存标记移动到相对于所述一个第二节点位于与第一方向相反的第二方向上的另一第二节点,其中,各第一节点与所述多个第二节点中的至少一个对应;以及比较单元,所述比较单元对于作为第一流水线的所述多个第一节点之一的关注节点,比较保持在关注节点中的地址与保持在关注节点所对应的第二节点中的至少一个中的高速缓存标记,以确定地址是否与高速缓存标记匹配,所述方法包括:在高速缓存存储器中存储被存储在外部存储器件中的数据中的一些;控制获取单元以从处理装置获取数据的请求;控制第一输入单元以复制获取的数据的地址并且将复制的地址输入到所述多个数据处

理装置的第一流水线 ;控制第二输入单元以将存储于高速缓存存储器中的数据的高速缓存标记输入到所述多个数据处理装置之一的第二流水线 ;以及控制确定单元以在确定比较单元的比较结果指示匹配时确定获取的数据的高速缓存命中,其中,所述多个数据处理装置的第二流水线被串行连接。

[0009] 从(参照附图)对示例性实施例的以下描述,本发明的进一步的特征将变得明显。

附图说明

- [0010] 图 1 是示出图像处理装置 100 的基本布置的示例性总体框图 ;
- [0011] 图 2 是示出高速缓存确定单元 120 的基本布置的示例性电路图 ;
- [0012] 图 3 是示出数据处理装置 200 的基本布置的示例性电路图 ;
- [0013] 图 4 是高速缓存确定单元 400 的示例性电路图 ;
- [0014] 图 5 是高速缓存确定单元 500 的示例性电路图 ;
- [0015] 图 6 是用于详细解释解码器 552 的示例性电路图 ;
- [0016] 图 7 示出分割 (partition) 信息表的示例性格式 ;
- [0017] 图 8 是示出分割信息表的实际例子的示例性示图 ;
- [0018] 图 9 是用于解释高速缓存存储器 900 的示例性示图 ;
- [0019] 图 10 是使用纵横开关 (crossbar switch) 时的示例性电路图 ;以及
- [0020] 图 11 是用于解释常规的逆流流水线技术的框图。

具体实施方式

[0021] 以下将参照附图描述本发明的实施例。

[0022] [图像处理装置 100]

[0023] 图 1 是示出包含高速缓存装置 110 的图像处理装置 100 的总体布置的例子的示例性框图。在高速缓存装置 110 中,数据处理装置(后面将描述)被应用作为完全相联的 (fully-associative) 高速缓存确定单元 120。

[0024] 图像处理装置 100 包含 CPU 160、作为外部存储器件的 DRAM165、DRAM 控制器 166 和系统总线 164。图像处理装置 100 还包含作为用于读出数据的 DMAC(Direct Memory Access Controller,直接存储器访问控制器)的 RDMAC 162 和用于写入数据的 WDMAC 163, 以从 / 向 DRAM 165 读 / 写数据。图像处理装置 100 进一步包含执行诸如图像处理的处理的处理装置 (P.A.)A 101 至 D 104、以及高速缓存装置 110。多个处理装置共享高速缓存装置 110。即,高速缓存装置 110 是共享的高速缓存装置。图像处理装置 100 可包含任意数量的处理装置,每个处理装置可通过执行快速固定 (permanent) 处理的流水线电路、或者通过虽然处理速度低但可灵活改变处理内容的处理器和程序来配置。

[0025] CPU 160 经由控制总线 161 控制 RDMAC 162、WDMAC 163 和处理装置 A 101 至 D 104,由此系统性地控制整个图像处理装置 100。依据来自 CPU 160 的指令,RDMAC 162 经由系统总线 164 和 DRAM 控制器 166 读出存储于 DRAM 165 中的图像数据,并且将读出的图像数据输入到处理装置 A 101。处理装置 A 101 执行希望的图像处理,并且将作为处理结果的图像数据输出到处理装置 B 102。处理装置 B102、C 103 和 D 104 如处理装置 A 101 中那样类似地执行各种类型的图像处理,并且将作为处理结果的图像数据输出到随后的处理

装置。处理装置 D 104 将作为处理结果的图像数据输出到 WDMAC 163。WDMAC 163 依据来自 CPU 160 的指令经由系统总线 164 和 DRAM 控制器 166 将从处理装置 D 104 接收的图像数据存储在 DRAM 165 中。图像处理装置 100 通过执行上述的系列的操作来实施图像处理。

[0026] 在上述的图像处理期间,处理装置 A 101 至 D 104 可经由共享的高速缓存装置 110 从 DRAM 165 读出各种需要的数据(设置、表、属性信息等),并且可在图像处理中使用读出的数据。当通过处理器和程序来配置处理装置中的任一个时,可经由高速缓存装置 110 依次读出程序以实施处理。

[0027] [高速缓存装置 110]

[0028] 以下将详细描述高速缓存装置 110 的布置和操作。上述的处理装置 A 101 至 D 104 在它们经由高速缓存装置 110 从 DRAM 165 读出数据时将 DRAM 165 上的数据的存储地址输出到 I/F 112。高速缓存装置 110 包含用于访问请求的环形总线 111,以传送 DRAM 165 上的数据的存储地址。与各自的处理装置对应的 I/F 112 经由该环形总线 111 相互连接。从各自的处理装置输出的存储地址 113 经由 I/F 112 和环形总线 111 被输入到高速缓存确定单元 120。高速缓存确定单元 120 基于输入的存储地址 113 确定高速缓存命中或遗漏(miss)。

[0029] [高速缓存确定单元 120]

[0030] 以下将参照图 2 详细描述高速缓存确定单元 120 的电路布置的例子。高速缓存确定单元 120 包含数据处理装置 200 和高速缓存确定装置 280。

[0031] 要输入到高速缓存确定单元 120(第一输入)的存储地址 113 包含:

[0032] ●地址有效(validity)信号“valid”;

[0033] ●地址信号“data”;和

[0034] ●处理装置 101 至 104 中的每一个的识别信号“ID”。

[0035] 这些信号被输入到数据处理装置 200。

[0036] [数据处理装置 200]

[0037] 以下将参照图 3 详细描述数据处理装置 200 的电路布置的例子。图 3 是关注于数据处理装置 200 的一部分的示图。第一流水线 300 具有多个节点(第一节点)。如果选择这些节点中的一个作为关注节点(例如,节点 310),那么数据元素(data element)(第一数据)从关注节点沿第一方向(图 1 中的右方向)移动到下游侧的下一第一节点(例如,节点 330)。如图 2 所示,第一流水线 300 从数据处理装置 200 外面接收:

[0038] ●地址有效信号“valid”;

[0039] ●地址信号“data”;

[0040] ●作为地址信号的处理结果的处理结果信号“tag_id”;和

[0041] ●处理装置的识别信号“ID”。

[0042] 信号“valid”、“data”和“ID”包含于存储地址 113 中,并且,信号“tag_id”例如为缺省值。

[0043] 反过来参照图 3,例如,节点 310 接收:

[0044] ●地址有效信号“valid[1-1]”312;

[0045] ●地址信号“data[1-1]”314;和

[0046] ●各处理装置的识别信号“ID[1-1]”。

[0047] 在图 3 中,为了简化,信号“valid”和“ID”由一条信号线表示。并且,数据处理电

路 320 接收：

[0048] ●地址信号“data[l-1]”314；和

[0049] ●作为地址信号的处理结果的处理结果信号“tag_id[l-1]”316。

[0050] 数据处理电路 320 对这些输入信号施加后面将描述的处理，并且将处理结果输出到节点 310。

[0051] 当作为驱动信号的信号“enable”121 有效（被断言（asserted））时，包含流水线寄存器的节点 310 使用从上游侧的节点和数据处理电路 320 输入的数据元素来更新目前为止存储的数据元素。另一方面，当信号“enable”121 无效（被去断言（deasserted））时，节点 310 忽略从上游侧输入的数据元素，使存储的数据元素保持原样，并且不更新数据元素。为了便于描述，在第一流水线 300 之中，包含一个节点和向该节点输入处理结果信号的数据处理电路的区段（section）将被称为级（stage）。特别地，在图 3 中，包含节点 310 和数据处理电路 320 的区段将被称为第一级。

[0052] 节点 310 向下游侧的节点 330 和数据处理电路 340 输出存储的数据，即：

[0053] ●地址有效信号“valid[l]”332；

[0054] ●地址信号“data[l]”334；

[0055] ●处理结果信号“tag_id[l]”336；和

[0056] ●识别信号“ID[l]”。

[0057] 在图 3 中，在第一流水线 300 之中，包含节点 330 和数据处理电路 340 的区段将被称为第二级。如第一级中那样，在第二级中执行各种类型的数据处理。处理之后的数据元素被暂时存储在节点 330 中。该存储操作与第一级中的存储操作相同。

[0058] 并且，节点 330 输出：

[0059] ●地址有效信号“valid[l+1]”352；

[0060] ●地址信号“data[l+1]”354；

[0061] ●处理结果信号“tag_id[l+1]”356；和

[0062] ●识别信号“ID[l+1]”。

[0063] 通过该操作，在第一流水线 300 中，数据元素（有效信号“valid”、地址信号“data”、处理结果信号“tag_id”和识别信号“ID”）从作为“上游侧”的图面上的左侧向作为“下游侧”的图面上的右侧移动。

[0064] 除了第一流水线 300 以外，数据处理装置 200 还包含第二流水线 360。该第二流水线 360 具有多个节点（第二节点），并且将保持在这些第二节点中的流水线寄存器中的数据元素移动到沿与第一流水线相反的方向（第二方向）定位的相邻节点。更具体而言，第二流水线 360 从作为“上游侧”的图面上的右侧向作为“下游侧”的图面上的左侧移动数据元素。如图 2 所示，第二流水线 360 从数据处理装置 200 外面接收地址有效信号“tag_valid”和地址信号“tag_data”（第二输入）。

[0065] 反过来参照图 3，节点 370 从上游侧接收地址有效信号“tag_valid[i+2]”372 和地址信号“tag_data[i+2]”374。在上述的数据处理电路 340 中使用这些输入信号。

[0066] 当驱动信号“shift”362 有效（被断言）时，节点 370 使用从上游侧的节点输入的数据元素来更新目前为止存储的数据元素。另一方面，当驱动信号“shift”362 无效（被去断言）时，节点 370 忽略从上游侧输入的数据元素，使存储的数据元素保持原样，并且不

更新数据元素。在第二流水线 360 之中,包含一个节点的区段将被称为级。特别地,在图 3 中,包含节点 370 的区段将被称为第一级。

[0067] 节点 370 将存储于流水线寄存器中的地址有效信号“tag_valid[i+1]”382 和地址信号“tag_data[i+1]”384 输出到节点 380 与数据处理电路 320 和 340。并且,节点 380 将有效信号“tag_valid[i]”392 和地址信号“tag_data[i]”394 输出到下游侧。通过该操作,在第二流水线 360 之中,作为数据元素的有效信号“tag_valid”和地址信号“tag_data”从作为“上游侧”的图面上的右侧移动到作为“下游侧”的图面上的左侧。

[0068] [数据处理电路]

[0069] 以下将详细描述数据处理电路 320 和 340。数据处理电路比较两个方向上的地址信号“data”和“tag_data”(第一比较)。当这些信号彼此相等时,数据处理电路将信号“tag_data”的存储标号(index)(上述的“[i]”、“[i+1]”或“[i+2]”)存储作为信号“tag_id”。然后,同步于“data”,信号“tag_id”作为第一流水线 300 的数据处理结果从作为“上游侧”的图面上的左侧移动到作为“下游侧”的图面上的右侧。以此方式,在信号“tag_id[1]”336 中,设定第二流水线 360 的数据元素的存储标号,其具有与第一流水线 300 的第 1 个数据元素“data[1]”334 的值相等的值。

[0070] 更具体而言,当有效信号“tag_valid[i]”392 有效时,数据处理电路 320 控制比较电路 322 以比较地址信号“data[l-1]”314 和“tag_data[i]”394。当作为比较结果这些信号彼此相等时,选择器 326 选择信号“tag_data[i]”394 的存储标号“Node = i”。该选择值在信号“tag_id[l-1]”316 中被设定作为第二流水线 360 的数据元素的存储标号,其具有与第一流水线 300 的信号“data[l-1]”314 的值相等的值。

[0071] 如上所述,当第一流水线 300 和第二流水线 360 同时操作时,在一些情况下可能出现比较差错。为了避免这种差错,当有效信号“tag_valid[i+1]”382 有效时,数据处理电路 320 控制比较电路 324 以比较地址信号“data[l-1]”314 和“tag_data[i+1]”384(第二比较)。当作为比较结果这些信号彼此相等时,选择器 326 优先选择信号“tag_data[i+1]”384 的存储标号“Node = i+1”。

[0072] 当作为两个比较结果两对比较信号不彼此相等时,选择器 326 选择输入的处理结果信号“tag_id[l-1]”316。当驱动信号“shift”362 有效(被断言)并且第二流水线 360 操作时,第二流水线 360 的数据元素向作为“下游侧”的图面上的左侧移动。因此,在这种情况下,各信号“tag_data”的存储标号指示左相邻的存储标号。由此,通过使用减量器(减法器)328 将选择器 326 的选择结果减“1”,调整存储标号。

[0073] 以下将补充存储标号选择方法。选择存储标号“Node = i”、“Node = i+1”和“tag_id[l-1]”316 中的一个。选择存储标号的准则可以是简单的准则,例如,“优先选择具有较大值的存储标号”。例如,以下将检查其中数据信号“data[l-1]”314 和“tag_data[i+1]”384 彼此相等并且驱动信号“shift”362 有效的情况。在这种情况下,为了避免上述的比较差错,选择“Node = i+1”是重要的,并且,该操作与方法“优先选择具有较大值的存储标号”匹配。另一方面,当驱动信号“shift”362 无效时,不需要避免上述的比较差错,并且,不需要选择“Node = i+1”。但是,不管“Node = i+1”的选择/不选择,作为第一流水线 300 的下游级的第二级都再次评价地址信号“data[1]”334 和“tag_data[i+1]”384 之间的比较。出于该原因,当驱动信号“shift”362 无效时,可以进行或者可以不进行第一级中的与信号

“tag_data[i+1]”384 的比较。相反,可以通过方法“优先选择具有较大值的存储标号”进行选择。在信号“tag_id[l-1]”316 中设定以此方式选择的值,所述值指示第二流水线 360 的数据元素的存储标号,其具有与信号“data[l-1]”314 的值相等的值。

[0074] 仅出于示例性的目的给出已参照图 3 解释的例子。例如,可以在选择器 326 中代入驱动信号“shift”362,以控制在驱动信号“shift”362 无效时不选择“Node = i+1”。当两个比较电路 322 和 324 的比较结果为<伪 (false)>时,选择输入数据信号“tag_id[l-1]”316。在任何情况下,当驱动信号“shift”362 有效时,执行将存储标号减“1”的调整以应对存储位置向“下游侧”的移动。出于该原因,不执行不选择“Node = i+1”的控制,并且,在选择“Node = i+1”之后再次执行将存储标号减“1”的调整。

[0075] 这同样适用于数据处理电路 340。当有效信号“tag_valid[i+1]”382 有效时,比较电路 342 比较地址信号“data[l]”334 和“tag_data[i+1]”384。当作为比较结果两个信号彼此相等时,选择器 346 选择信号“tag_data[i+1]”384 的存储标号“Node = i+1”。为了准备第二流水线 360 的操作,当有效信号“tag_valid[i+2]”372 有效时,数据处理电路 340 控制比较电路 344 以比较数据信号“data[l]”334 和“tag_data[i+2]”374。当作为比较结果这些信号彼此相等时,选择器 346 优先选择信号“tag_data[i+2]”374 的存储标号“Node = i+2”。当作为两个比较结果两对比较信号不彼此相等时,选择器 346 选择处理结果信号“tag_id[l]”336。当驱动信号“shift”362 有效并且第二流水线 360 因此操作时,信号“tag_data”的存储标号向作为“下游侧”的图面上的左侧移动“1”。出于该原因,从选择器 346 的选择结果使用减量器(减法器)348 将存储标号减“1”。以此方式,可以调整匹配检测结果,并且,可以通过简单的处理来实现更精确并且更快速的数据处理。如上所述,数据处理装置 200 可在流水线的各自的级中确实地且快速地比较沿相反方向移动的数据元素。

[0076] 上述的数据处理电路 320 和 340 中的每一个具有第二流水线 360 中要比较的数据元素针对第一流水线 300 的每一个数据元素的比较电路。并且,数据处理电路 320 和 340 中的每一个在第二流水线 360 操作的假定下具有第二流水线 360 的“上游”数据元素对于第二流水线 360 的要比较的数据元素的新的比较电路。由于这些电路可避免可能在日本专利 No. 3588487 的布置中出现的对于各特定级的联锁(interlocking)(内部停止)的问题,因此,可以总是获得高的处理性能而不停止数据处理。

[0077] 在数据处理装置 200 中,为了计算两个数据元素彼此相等时的存储位置,数据处理电路 320 和 340 中的每一个在第二流水线 360 操作的假定下包含用于将处理结果减“1”的减量器。然后,第二流水线的数据元素的存储标号被事先分配,以从第二流水线的“下游侧”向“上游侧”如“0、1、2、...、i、i+1、i+2、...、N-1 (i 和 N 是正值, $i < N$)”那样增加。将在后面描述这种分配的优点。当以相反的次序分配存储标号时,减量器自然用作被用于将存储标号加“1”的增量器。

[0078] 以下将再次参照图 2 描述数据处理装置 200 的布置。在数据处理装置 200 中,第一流水线 300 和第二流水线 360 均包含八个级。图 2 中的 Data Slot[0] 至 Data Slot[7] 与图 3 所示的第一流水线 300 的节点 310 和 330 对应。图 2 中的 Tag Slot[0] 至 Tag Slot[7] 与图 3 所示的第二流水线 360 的节点 370 和 380 对应。Judge[0] 至 Judge[7] 与图 3 所示的数据处理电路 320 和 340 对应。如上所述,通过设置多个级,数据处理装置 200 可通过流

流水线操作来对许多数据元素进行并行分布和比较。

[0079] 存储地址 113 通过数据处理装置 200 的第一流水线 300 的 Data Slot 移动。数据处理装置 200 可同时存储八个信号“tag_data”，并且高速缓存标记 (tag) 信息被存储于这八个信号“tag_data”中的每一个中。使用数据处理装置 200 的例子的高速缓存装置与 8 节点、完全相联的高速缓存装置对应。并且，从第二流水线 360 的“下游侧”依次串行设定第 0 个到第 7 个存储位置，并且，第二流水线 360 具有当驱动信号“shift”362 有效（被断言）时向“下游侧”移动数据元素的移动结构。通过该移动结构，最旧的高速缓存标记被存储于第 0 个存储位置处的信号“tag_data”中，最新的高速缓存标记被存储于第 7 个存储位置处的信号“tag_data”中。每次出现高速缓存遗漏时，高速缓存标记从第 7 个存储位置处的信号“tag_data”依次移动到第 0 个存储位置处的信号“tag_data”，并然后从第二流水线 360 被扫出 (sweep out)。尽管高速缓存确定单元 120 具有非常简单的结构，但其总是从最旧的高速缓存标记和高速缓存数据依次舍弃高速缓存标记和高速缓存数据。结果，不需要执行一般的高速缓存机构的复杂替换 (replace) 控制。

[0080] [高速缓存命中确定]

[0081] 以下将描述高速缓存命中确定序列。“高速缓存命中确定”是要确定由存储地址 113 指定的数据是否被存储于高速缓存存储器 190 中。当指定的数据被存储于高速缓存存储器 190 中时，确定高速缓存命中；否则，确定高速缓存遗漏。通过高速缓存确定装置 280 进行高速缓存命中确定。高速缓存确定装置 280 通过检查从数据处理装置 200 输出的处理结果信号“tag_id”(2 的补码表示 (complement representation)) 的具有 1 位长度的符号位，确定高速缓存命中或遗漏。当作为来自数据处理装置 200 的输出的有效信号“valid”有效（被断言）并且其符号位是“1”时，信号“tag_id”取负值，并且，高速缓存确定装置 280 确定高速缓存遗漏。当有效信号“valid”有效（被断言）并且其符号位是“0”时，信号“tag_id”取非负值，并且，高速缓存确定装置 280 确定高速缓存命中。当有效信号“valid”无效（被去断言）时，高速缓存确定装置 280 跳过 (skip) 确定。并且，当来自访问仲裁 (arbitration) 单元 130 的信号“enable”无效（被去断言）时，高速缓存确定装置 280 跳过确定。

[0082] 作为数据处理装置 200 的输出的地址信号“data”、有效信号“valid”和识别信号“ID”同步于信号“tag_id”被输入到高速缓存确定装置 280。当通过以上的处理确定高速缓存遗漏时，作为高速缓存遗漏时的地址“cache_miss_address”从高速缓存确定装置 280 输出该地址信号“data”。在高速缓存遗漏的情况下，高速缓存确定装置 280 使驱动信号“shift”362 有效（断言），并且作为高速缓存标记“tag_data”输出高速缓存遗漏时的地址“cache_miss_address”。每次高速缓存遗漏被确定，都使驱动信号“shift”362 有效，并且，如上所述处理结果“tag_id”被递减 (decrement)。

[0083] 即使当处理结果“tag_id”初始保持正值时，如果高速缓存遗漏被重复，那么第二流水线 360 移动，并且，常常从第二流水线 360 扫出高速缓存标记“tag_data”。如从以上的描述可以看出的那样，由于最旧的高速缓存标记的存储标号为零，因此扫出的处理结果“tag_id”取负值。这是仅需要在上述的高速缓存确定中检查信号“tag_id”的符号的原因。即，分配存储位置的序号 (number) 的方法被设计，使得最旧的高速缓存标记与第 0 个对应并且最新的高速缓存标记与第 (N-1) 个对应。结果，可通过确定第一流水线 300 的最后级

中的数据处理结果的符号,获得高速缓存命中确定。因此,可以非常简单地完成高速缓存确定。当从第二流水线 360 的“上游侧”向“下游侧”从第 0 个到第 (N-1) 个分配存储标号时,可通过检查信号“tag_id”的值是否比作为第二流水线的元素的数量 N 小,获得高速缓存命中确定。

[0084] 对于如上所述总是从最旧的高速缓存数据依次舍弃高速缓存数据的机构,高速缓存存储器 190 可使用环形 FIFO。在这种情况下,高速缓存确定单元 120 和高速缓存存储器 190 可容易地被同步化。注意,当高速缓存确定指示高速缓存命中时,希望的高速缓存数据被存储于由“tag_id”指示的位置处的高速缓存存储器中。

[0085] 通过上述的处理,高速缓存确定单元 120 向访问仲裁单元 130 输出以下的内容作为基于输入存储地址 113 的确定结果 125:

[0086] ●地址有效信号“valid”,

[0087] ●作为高速缓存遗漏时的 DRAM 165 的数据存储地址的地址信号“cache_miss_address”,

[0088] ●处理装置的识别信号“ID”;

[0089] ●作为高速缓存数据的存储地址的信号“tag_id”,和

[0090] ●高速缓存确定结果“cache_miss_flag”。

[0091] [访问仲裁单元]130

[0092] 以下将参照图 1 描述访问仲裁单元 130 的操作。当确定结果 125 中的有效信号“valid”有效(被断言)时,访问仲裁单元 130 操作;否则,它等待。访问仲裁单元 130 根据确定结果 125 中的高速缓存确定结果“cache_miss_flag”的有效(被断言)/无效(被去断言)状态,执行以下的处理。

[0093] ●当高速缓存确定结果“cache_miss_flag”有效时,访问仲裁单元 130 评价发送队列 150、接收队列 170 和等待队列 140 的各自存储区域的空状态。当三个队列均具有空区域时,访问仲裁单元 130 使信号“ID、tag_id、chche_miss_flag”135 在等待队列 140 中入列(enqueue)。同时,访问仲裁单元 130 使地址信号“cache_miss_address”132 在发送队列 150 中入列。如果不存在空区域,那么访问仲裁单元 130 使驱动信号“enable”121 无效(去断言)以停止高速缓存确定单元 120,并且进行等待直到存储区域变为空。

[0094] ●当高速缓存确定结果“cache_miss_flag”无效时,访问仲裁单元 130 评价等待队列 140 的空状态。如果存在空区域,那么访问仲裁单元 130 使信号“ID、tag_id、cache_miss_flag”135 在等待队列 140 中入列。如果不存在空区域,那么访问仲裁单元 130 使驱动信号“enable”121 无效(去断言)以停止高速缓存确定单元 120,并且进行等待直到存储区域变为空。

[0095] [高速缓存存储器仲裁单元 180]

[0096] 以下将描述高速缓存存储器仲裁单元 180 的操作。高速缓存存储器仲裁单元 180 评价是否在接收队列 170 和等待队列 140 的存储区域中保持数据。高速缓存存储器仲裁单元 180 使要被处理的高速缓存确定结果“ID、tag_id、cache_miss_flag”从等待队列 140 出列(dequeue)。当等待队列 140 为空时,由于不存在要被处理的高速缓存确定结果,因此高速缓存存储器仲裁单元 180 进行等待而没有任何处理。接下来,高速缓存存储器仲裁单元 180 取决于高速缓存确定结果“cache_miss_flag”是无效(被去断言)还是有效(被断言)

而执行以下的处理。

[0097] ●当高速缓存确定结果“cache_miss_flag”无效（被去断言）时，这意味着高速缓存命中。由此，高速缓存存储器仲裁单元 180 基于从等待队列 140 出列的数据元素中的“tag_id”和 FIFO 高速缓存存储器 190 的写入指针，计算高速缓存存储器 190 中的存储地址。高速缓存存储器仲裁单元 180 从高速缓存存储器 190 基于该存储地址读出高速缓存的数据作为数据“read_data”192。然后，高速缓存存储器仲裁单元 180 将高速缓存数据“valid、ID、cache_data”185 输出到 I/F 116。

[0098] ●当高速缓存确定结果“cache_miss_flag”有效（被断言）时，这意味着高速缓存遗漏。高速缓存存储器仲裁单元 180 确认是否在接收队列 170 中从 DRAM 165 接收到未存储于高速缓存存储器 190 中的数据。如果没有接收到数据，那么高速缓存存储器仲裁单元 180 进行等待，直到接收到数据。如果接收到数据，那么高速缓存存储器仲裁单元 180 使要被更新的高速缓存数据从接收队列 170 出列。然后，高速缓存存储器仲裁单元 180 在由写入指针所指示的高速缓存存储器 190 的存储区域中写入出列的数据作为数据“write_data”182。同时，高速缓存存储器仲裁单元 180 将高速缓存数据“valid、ID、cache_data”185 输出到 I/F 116。接下来，高速缓存存储器仲裁单元 180 使高速缓存存储器 190 的写入指针递增（increment）。当超过 FIFO 容量时，高速缓存存储器仲裁单元 180 将写入指针复位为零。

[0099] 高速缓存装置 110 将通过上述的处理所获得的高速缓存数据分发给处理装置 101 至 104。高速缓存装置 110 包含用于分发高速缓存数据的环形总线 115。与这些处理装置对应的 I/F 116 经由该环形总线 115 相互连接。与各自的处理装置连接的 I/F 116 中的每一个接收作为来自高速缓存存储器仲裁单元 180 的输出的高速缓存数据“valid、ID、cache_data”185。然后，当接收的高速缓存数据的识别信号“ID”与连接的处理装置的识别信号“ID”匹配时，相应的 I/F 116 将高速缓存数据输出到处理装置。如果两个识别信号不匹配，那么各 I/F 116 经由环形总线 115 向位于随后的位置处的另一 I/F 116 发送接收的高速缓存数据。

[0100] 高速缓存装置 110 采用非阻挡高速缓存机构，以隐藏作为高速缓存遗漏时的处罚的重填等待时间（refill latency）。即使当确定结果 125 指示高速缓存遗漏时，该机构也在等待队列 140 中保存后面需要的信息“ID、tag_id、cache_miss_flag”135。然后，高速缓存装置 110 在完成从 DRAM 165 读出高速缓存遗漏数据并且将其存储于高速缓存存储器 190 中的处理之前，执行下一像素的高速缓存确定处理。通过该处理，即使在从 DRAM 165 向高速缓存存储器 190 重填高速缓存遗漏数据时，高速缓存装置 110 也可对于随后的像素执行高速缓存确定。因此，可以抑制高速缓存遗漏时的性能劣化。

[0101] 高速缓存装置 110 可使用非常简单的机构来实现可被多个处理装置共享的完全相联的高速缓存装置。当多个处理装置共享一个高速缓存装置 110 时，具有低的相关性的地址被接连输入到高速缓存确定单元 120。由于采用一般直接映射方法的高速缓存确定单元从各地址的低位（lower bits）计算用于管理高速缓存标记的标记存储器的存储地址，因此，具有低的相关性的这种地址容易导致高速缓存冲突。与处理装置的数量对应地增加组相联（set-associative）节点的数量是减少高速缓存冲突可能性的一种解决方案。但是，当处理装置的数量变得非常大时，必须应对非常大量的节点。由此，由于高速缓存确定单元中的选择器的逻辑级的数量大，因此一般的高速缓存装置实现方法变得难以使定时收敛

(converge),并且不能以高的操作频率来操作高速缓存确定单元。作为对比,由于高速缓存确定单元 120 通过流水线配置来获得确定,因此它可以以非常高的操作频率操作。并且,由于从较旧的数据依次自动删除数据,因此高速缓存确定单元 120 不需要现有技术中所需要的高速缓存冲突时的任何复杂的替换控制。出于该原因,高速缓存确定单元 120 和高速缓存存储器仲裁单元 180 可通过非常简单的机构被同步化,并且,可以使用 FIFO 作为高速缓存存储器。由此,这种布置对于提高高速缓存装置 110 的操作频率是有利的。更重要的是,高速缓存装置 110 采用完全相联方法,并且,决不导致由于地址的低位相同所造成的高速缓存冲突。

[0102] 将描述使用图 4 所示的高速缓存确定单元 400 的高速缓存装置。图 4 是用于解释高速缓存确定单元 400 的电路布置的例子的电路图。代替图 1 所示的高速缓存装置 110 的高速缓存确定单元 120,使用高速缓存确定单元 400。与上述的高速缓存装置 110 的基本布置中相同的部分的描述将不被重复。高速缓存确定单元 400 包含复制装置 410、运算装置 430 和高速缓存确定装置 480。运算装置 430 包含多个数据处理装置 (D. P. A.) 200_1 至 200_N,这些数据处理装置中的每一个具有与使用图 2 描述的数据处理装置 200 相同的布置。以此方式,由于高速缓存确定单元 400 包含多个上述的数据处理装置,因此增加了可同时比较的数据元素的数量。由于图 4 所示的高速缓存确定单元 400 包含多个基本的数据处理装置,因此可以再利用已设计和验证的数据处理装置,由此提高开发效率。

[0103] 包含于高速缓存确定单元 400 中的各自的 Data Slot 与图 3 所示的节点 310 和 330 相同,并且当信号“enable”121 有效(被断言)时使用新的数据元素更新目前为止存储的数据元素。节点 411 从 I/F 112 接收存储地址 113。如上所述,存储地址 113 包含有效信号“valid”、地址信号“data”和识别信号“ID”。复制装置 410 如 420 指示的那样复制存储地址 113,以在节点 412_1 至 412_N 中将其重定时(retime)。节点 412_1 至 412_N 分别将存储的数据元素输出到相应的数据处理装置 200_1 至 200_N 的第一流水线 300。以此方式,同一存储地址 113 的复制品在相同的定时被并行输入到各自的数据处理装置的第一流水线 300。

[0104] 运算装置 430 接收从高速缓存确定装置 480 的解码器 452 输出的信号“cache_miss_flag”444 作为输入到各自的数据处理装置 200_1 至 200_N 的驱动信号“shift”。然后,所有的数据处理装置 200_1 至 200_N 的第二流水线 360 的数据元素同步移动。从高速缓存确定装置 480 输出的信号“cache_miss_flag”和“cache_miss_address”作为信号“tag_valid”和“tag_data”被输入到数据处理装置 (1) 200_1 的第二流水线 360。从数据处理装置 (1) 的第二流水线 360 输出的信号“tag_valid”和“tag_data”作为信号“tag_valid”和“tag_data”被输入到数据处理装置 (2) 200_2 的第二流水线 360 (440_1)。这同样适用于随后的数据处理装置 (440_2、440_N)。从数据处理装置 (i) 的第二流水线 360 输出的信号“tag_valid”和“tag_data”作为信号“tag_valid”和“tag_data”被输入到数据处理装置 (i+1) 的第二流水线 360。以此方式,在运算装置 430 中,数据处理装置被顺次连接,使得某数据处理装置的第二流水线的输出“tag_valid、tag_data”与随后的数据处理装置的第二流水线的输入串行连接。

[0105] 高速缓存确定装置 480 从所有的数据处理装置 200_1 至 200_N 接收处理结果“tag_id(1)、tag_id(2)、...、tag_id(N)”450。高速缓存确定装置 480 进一步接收从数据

处理装置 200_1 至 200_N 中的任一个的第一流水线 300 输出的信号“valid、data”。解码器 452 确认有效信号“valid”453 是否有效（被断言）以及信号“enable”121 是否有效。如果两个信号均有效，那么高速缓存确定装置 480 控制解码器 452 以确认 N 个处理结果“tag_id(1)、tag_id(2)、...、tag_id(N)”450 的符号位 451。如果符号位 451 包含至少一个“0”，那么作为高速缓存确定结果解码器 452 确定高速缓存命中，并且输出“0”作为信号“cache_miss_flag”444。如果所有的符号位 451 为“1”，那么作为高速缓存确定结果解码器 452 确定高速缓存遗漏，并且输出“1”作为信号“cache_miss_flag”444。在高速缓存命中的情况下，解码器 452 将与符号位的“0”对应的数据处理装置的序号 456 输出到选择器 454。选择器 454 从处理结果“tag_id(1)、tag_id(2)、...、tag_id(N)”450 选择由序号 456 指示的数据处理装置的“tag_id”，并且作为信号“tag_id”458 将其输出。当信号“enable”121 有效（被断言）时，节点 460 使用输入的数据元素更新目前为止存储的数据元素。然后，节点 460 将由自身存储的数据元素作为确定结果 125 输出到访问仲裁单元 130。

[0106] 如上所述，在使用图 4 所示的高速缓存确定单元 400 的高速缓存装置中，单个地址信号“data”在各自的数据处理装置中同时与多个不同的地址信号“tag_valid”比较。于是，可以容易地实现具有更大量的节点的完全相联的高速缓存装置。使用高速缓存确定单元 400 的高速缓存装置包含各具有基本布置的多个数据处理装置，并且可通过仅连接各自的数据处理装置的第一流水线 300 和第二流水线 360 而实现。出于该原因，可以再利用已设计并验证的数据处理装置。结果，例如，在需要增加 / 减少完全相联的高速缓存装置的节点的数量开发中，可以提高开发效率。并且，高速缓存确定单元 400 使用流水线配置执行数据处理，并且可通过仅计算所有符号位的逻辑积来确定高速缓存遗漏。以此方式，高速缓存装置可以以与常规的高速缓存装置相比非常高的操作频率操作。

[0107] 注意，在高速缓存遗漏的情况下，可作为位于最后的位置处的数据处理装置 (N) 200_N 的第二流水线 360 的输出来扫出数据元素“tag_data、tag_valid”。这些扫出的数据元素被输出到节点 460，节点 460 可将作为“sweep_address、sweep_address_valid”包含这些数据元素的确定结果 125 输出到访问仲裁单元 130。使用这些数据元素，可以实现回写 (write-back) 高速缓存装置。

[0108] 在回写高速缓存装置中，暂时存储于高速缓存存储器 190 中的高速缓存数据被通过处理装置 A 101 至 D 104 写出的写入数据替换。出于该原因，暂时存储于高速缓存存储器 190 中的数据必须被保存于作为外部存储器的 DRAM 165 中。为了应对回写高速缓存装置并允许处理装置 A 101 至 D 104 执行写入操作，要输出的存储地址 113 包含：

[0109] ● 写入标记“write_enable”；和

[0110] ● 写入数据“write_data”。

[0111] 当写入标记“write_enable”有效（被断言）时，写入数据“write_data”在高速缓存存储器 190 的存储地址处被写入。当写入标记“write_enable”无效（被去断言）时，执行上述的读取高速缓存装置的操作。然后，在高速缓存确定单元 400 中的各自的数据处理装置 200_1 至 200_N 的第一流水线 300 中同步于包含于存储地址 113 中的其它的数据元素而移动数据元素“write_enable”和“write_data”。数据处理装置 (N) 200_N 将数据元素“write_enable”和“write_data”同步于其它的数据元素而输出到节点 460。节点 460 将包含这些数据的确定结果 125 输出到访问仲裁单元 130。即，该确定结果 125 包含：

- [0112] ● 作为高速缓存遗漏时的 DRAM 的数据存储地址的地址信号“cache_miss_address”;
- [0113] ● 处理装置的识别信号“ID”;
- [0114] ● 高速缓存数据的存储地址“tag_id”;
- [0115] ● 高速缓存确定结果“cache_miss_flag”;
- [0116] ● 写入标记“write_enable”;
- [0117] ● 写入数据“write_data”;
- [0118] ● 扫出的地址信号“sweep_address”;和
- [0119] ● 扫出的地址有效信号“sweep_address_valid”。

[0120] 确定结果 125 经由等待队列 140 到达高速缓存存储器仲裁单元 180。高速缓存存储器仲裁单元 180 以与如上所述相同的方式使用“ID、tag_id”计算高速缓存存储器 190 的存储地址。当高速缓存确定结果“cache_miss_flag”无效（被去断言）时，由于这意味着高速缓存命中，因此高速缓存存储器仲裁单元 180 将写入数据“write_data”写入在高速缓存存储器 190 的存储地址处。另一方面，当高速缓存确定结果“cache_miss_flag”有效（被断言）时，由于这意味着高速缓存遗漏，因此高速缓存存储器仲裁单元 180 不能立即将写入数据“write_data”写入在高速缓存存储器 190 中。当扫出的地址有效信号“sweep_address_valid”有效（被断言）时，暂时存储于高速缓存存储器 190 的存储地址处的数据被重写（overwrite）。由此，高速缓存存储器仲裁单元 180 从高速缓存存储器 190 读出存储地址处的高速缓存数据，并且在由扫出的地址信号“sweep_address”指示的 DRAM 165 的区域中保存该高速缓存数据。之后，高速缓存存储器仲裁单元 180 在高速缓存存储器 190 中重填由地址信号“cache_miss_address”指示的作为外部存储器的 DRAM 165 的区域中的数据。此时，高速缓存存储器仲裁单元 180 通过写入数据“write_data”重写从 DRAM 165 读出的数据，并且在高速缓存存储器 190 的存储地址处暂时存储重写的的数据。当读出的数据和“write_data”具有相同的大小（相同的字节数）时，高速缓存存储器仲裁单元 180 可跳过该重填操作。当扫出的地址有效信号“sweep_address_valid”无效（被去断言）时，不在高速缓存存储器 190 的存储地址处暂时存储有效的高速缓存数据。因此，高速缓存存储器仲裁单元 180 不需要在高速缓存存储器 190 的存储地址处保存暂时存储的高速缓存数据。在这种情况下，如果必要的话，高速缓存存储器仲裁单元 180 可在执行上述的重填操作的同时在高速缓存存储器 190 的存储地址处暂时存储数据“write_data”作为高速缓存数据。

[0121] 通过上述的布置，可以容易地实现回写高速缓存装置。用作回写高速缓存装置的高速缓存装置的高速缓存确定单元 400 也具有与读取高速缓存装置相同的优点。特别地，从较旧的数据依次自动删除数据，并且，不需要执行高速缓存冲突时的复杂替换控制。于是，仅需要在作为外部存储器的 DRAM 165 中从较旧的数据依次保存暂时存储于高速缓存存储器中的高速缓存数据。如上所述，可通过非常简单的方法来实现回写高速缓存装置所特有的回写控制。

[0122] 以下将描述使用图 5 所示的高速缓存确定单元 500 的高速缓存装置。图 5 是用于解释包含分割信息表的高速缓存确定单元 500 的电路布置的示例性电路图。替代图 1 所示的高速缓存装置 110 的高速缓存确定单元 120，使用高速缓存确定单元 500。

[0123] 高速缓存确定单元 500 包含复制装置 510、运算装置 530 和高速缓存确定装置 580。运算装置 530 包含多个数据处理装置 200_1 至 200_N, 这些数据处理装置中的每一个具有与使用图 2 描述的数据处理装置 200 相同的布置。以此方式, 如高速缓存确定单元 400 中那样, 高速缓存确定单元 500 包含多个上述的数据处理装置, 由此增加了可同时比较的数据元素的数量。并且, 当多个处理装置 A 101 至 D 104 共享高速缓存装置 110 时, 高速缓存确定单元 500 可对于各自的处理装置切换要被各处理装置专用的高速缓存容量 (节点的数量)。通过切换各处理装置的专用分配, 可以给具有高的优先级的图像处理分配较大的高速缓存容量 (节点的数量)。即, 在执行图像处理时, 根据图像处理的目的来适应性地切换高速缓存容量 (节点的数量) 的专用分配, 由此以希望的效率执行图像处理。

[0124] 与复制装置 410 中不同, 图 5 所示的复制装置 510 包含分割信息表 515。因此, 将不给出与复制装置 410 中的部分共同的部分的描述, 并且, 以下将描述分割信息表 515。在分割信息表 515 中, CPU 160 事先设定分割信息, 以切换要被各自的处理装置专用的高速缓存容量 (节点的数量) 的分配。分割信息表示要被各自的处理装置专用的数据处理装置 200_1 至 200_N 的分配 (组)。以下将参照图 7 和图 8 详细描述分割信息表 515。分割信息表 515 如以图 7 所示的格式 700 那样允许对于处理装置的各识别信号 ID 702 设定分割信息 704。当存在 M 个处理装置并且在运算装置 530 中存在 N 个数据处理装置 200 时, 对应于识别信号 ID 702 = 1 至 M 中的每一个存储 N 位分割信息 704。N 位分割信息 704 从最高有效位 (the most significant bit) 依次用作标记, 该标记的每个位指示是否要使用由该位指示的数据处理装置 200。当各位为“1”时, 它指示所指示的数据处理装置的 < 使用 >; 当它是“0”时, 它指示 < 不使用 >。

[0125] 以下将参照图 8 描述分割信息表 515 的实际例子 800。在实际例子 800 中, 对应于四个识别信号 ID 702 = 1 至 4 存储四条 8 位分割信息 704。这四个识别信号 ID 702 分别与图 1 所示的四个处理装置 A 101 至 D 104 相关联。然后, 8 位分割信息 704 的各单个位从最高有效位依次与图 5 所示的数据处理装置 (1) 200_1 至 (N) 200_N 中的每一个相关联。例如, 处理装置 A 101 的分割信息为“10000000”, 并且具有指示处理装置 A 101 使用数据处理装置 (1) (在 820 处示出)、但它不使用剩余的数据处理装置 (2) 至 (8) 的设置。处理装置 B 102 的分割信息为“01110000”, 并且具有指示处理装置 B 102 使用数据处理装置 (2) 至 (4) (在 822 处示出)、但它不使用剩余的数据处理装置 (1) 和 (5) 至 (8) 的设置。同样地, 处理装置 C 103 的设置指示: 它使用数据处理装置 (5) 和 (7), 但它不使用剩余的数据处理装置。处理装置 D 104 的设置指示: 它使用数据处理装置 (6) 和 (8), 但它不使用剩余的数据处理装置。在该实际例子 800 中, 处理装置 A101 专用数据处理装置 (1), 处理装置 B 102 专用数据处理装置 (2) 至 (4), 处理装置 C 103 专用数据处理装置 (5) 和 (7), 并且, 处理装置 D 104 专用数据处理装置 (6) 和 (8)。各条分割信息 704 的设置仅需要在多个处理装置之间是排他的, 并且, 可以像处理装置 C103 和 D 104 的设置中那样不连续地分配数据处理装置 (在 824 处示出)。

[0126] 回过来参照图 5, 复制装置 510 使用输入的识别信号“ID”从分割信息表 515 读出与该识别信号“ID”对应的分割信息“partition”518, 并且将读出的信息输入到节点 511。节点 511 复制输入有效信号“valid”和输入数据信号“data、ID、partition”, 并且将它们输出到节点 512_1 至 512_N。节点 512_1 至 512_N 分别将由它们自身存储的数据元素输出

到相应的数据处理装置 200_1 至 200_N 的第一流水线 300。以此方式,复制装置 510 复制如数据处理装置 200 的数量那样多的数据元素以将它们重定时。

[0127] [第二流水线 360 的连接切换]

[0128] 在运算装置 530 中,如虚线部分 540 所示,连接各自的数据处理装置的第二流水线 360。根据从数据处理装置 (N) 200_N 输出的分割信息“partition”535 自动切换第二流水线 360 的连接。运算装置 530 分别计算从解码器 552 输出的信号“cache_miss_flag”559 和从数据处理装置 (N) 输出的分割信息“partition”535 的与数据处理装置 (1) 至 (N) 对应的各自位的逻辑积。然后,运算装置 530 输入计算的逻辑积作为相应的数据处理装置的驱动信号“shift”(544_1、544_2、...、544_N)。结果,所有数据处理装置之中的仅仅与分割信息中的 <使用> 对应的数据处理装置的第二流水线 360 的数据元素被同步移动。

[0129] 接下来,运算装置 530 对切换第二流水线 360 的连接所需要的数据元素进行汇编 (compile)。从各自的数据处理装置 (1) 至 (N) 的第二流水线 360 输出的数据元素 532_1 至 532_N、“cache_miss_flag”559 和“cache_miss_address(data)”555 被汇聚 (bundle) 于总线 542 上。然后,数据从总线 542 被输入到用于选择对于数据处理装置 (1) 至 (N) 的第二流水线 360 的输入的选择器 545_1、545_2、...、545_N。选择器 545_1、545_2、...、545_N 基于分割信息“partition”535 选择相应的数据元素,并且将选择的数据元素输入到数据处理装置 (1) 至 (N) 的第二流水线 360。并且,可以添加用于输出在高速缓存遗漏时被扫出的数据元素的选择器 548。扫出的数据元素基于分割信息“partition”535 从总线 542 被选择,并且作为数据元素“sweep_address、sweep_address_valid”被输出。如上所述,当高速缓存装置 110 被用作回写高速缓存装置时,使用这些数据元素。

[0130] 例如,实际例子 800 中对于第二流水线的输入选择方法如下。如从实际例子 800 可以看出的那样,当从数据处理装置 (N) 的第一流水线 300 输出的识别序号“ID”是“1”时,分割信息包含设置“10000000”。当与“ID”=“1”的数据元素相关联的高速缓存确定结果是高速缓存遗漏时,数据处理装置 (1) 的第二流水线 360 被移动。由此,作为对于数据处理装置 (1) 的第二流水线的输入,从总线 542 上的那些数据选择数据“cache_miss_flag”559 和“cache_miss_address(data)”555。作为扫出的数据元素“sweep_address、sweep_address_valid”,选择来自数据处理装置 (1) 的第二流水线 360 的数据元素 532_1。另一方面,如从实际例子 800 可以看出的那样,当从数据处理装置 (N) 的第一流水线 300 输出的识别序号“ID”是“2”时,分割信息包含设置“01110000”。当与“ID”=“2”的数据元素相关联的高速缓存确定结果是高速缓存遗漏时,数据处理装置 (2) 至 (4) 的第二流水线被移动。由此,作为对于数据处理装置 (2) 的第二流水线 360 的输入,从总线 542 上的那些数据选择数据“cache_miss_flag”559 和“cache_miss_address(data)”555。作为对于数据处理装置 (3) 的第二流水线的输入,来自数据处理装置 (2) 的第二流水线的输出 532_2 被选择。作为对于数据处理装置 (4) 的第二流水线的输入,来自数据处理装置 (3) 的第二流水线的输出被选择。并且,作为扫出的数据元素“sweep_address、sweep_address_valid”,来自数据处理装置 (4) 的第二流水线的输出被选择。这同样适用于与“ID”=“3”和“4”的数据元素相关联的第二流水线的连接的切换。即使当分割信息像实际例子 800 的部分 824 中那样包含不连续的设置时,也以数据处理装置的装置序号的升序依次连接第二流水线。即,来自要使用的数据处理装置的第二流水线的输出仅需要根据分割信息“partition”535 的设置与要使

用的下一数据处理装置的第二流水线的输入串行连接。

[0131] 高速缓存确定装置 580 接收所有数据处理装置 200_1 至 200_N 的处理结果“tag_id(1)、tag_id(2)、...、tag_id(N)”550。高速缓存确定装置 580 从数据处理装置 200_1 至 200_N 中的任一个接收信号“valid、data、ID、partition”。在图 5 中,从数据处理装置 (N) 200_N 输入这些信号,但是可从另一数据处理装置输入它们。与解码器 452 不同,解码器 552 接收分割信息“partition”535。以下将参照图 6 详细描述解码器 552。解码器 552 计算以下的四个信号 (1 位信号) 的逻辑积:

[0132] ●有效信号“valid”553;

[0133] ●通过反转 (invert) 数据处理装置 (1) 至 (N) 中的一个的处理结果“tag_id”的符号位 (反转符号位 551 的任意的 1 位) 获得的信号;

[0134] ●分割信息“partition”535 的与数据处理装置 (1) 至 (N) 中的一个对应的 1 位信号;和

[0135] ●驱动信号“enable”121。

[0136] 当逻辑积中的至少一个是“1”时,解码器 552 将高速缓存命中确定为高速缓存确定结果;当所有的逻辑积为“0”时,它将高速缓存遗漏确定为高速缓存确定结果。当所有的逻辑积为“0”时,解码器 552 使信号“cache_miss_flag”559 有效 (断言) 并且输出该信号。在高速缓存命中的情况下,解码器 552 将与逻辑积 = “1”对应的数据处理装置的装置序号 (1 至 N 中的任一个) 556 输出到选择器 554。高速缓存确定装置 580 使用选择器 554 基于高速缓存命中装置序号 556 从多个处理结果“tag_id(1) 至 tag_id(N)”550 选择高速缓存命中时的处理结果“tag_id”558,并且将其输出到节点 560。

[0137] 以下将描述使用高速缓存确定单元 500 的高速缓存装置的高速缓存存储器仲裁单元 180 的操作和高速缓存存储器 900 的布置。首先将描述图 9 所示的高速缓存存储器 900。替代图 1 所示的高速缓存存储器 190,使用高速缓存存储器 900。高速缓存存储器 900 也由 FIFO 配置。高速缓存存储器 900 被分成与数据处理装置 (1) 至 (N) 对应的区域。根据分割信息的设置,对于各 ID 确定作为 FIFO 的基准的基本指针和各区域的行数 (number of lines)。即使当区域像实际例子 800 的部分 824 中那样是不连续的时,也通过上述的第二流水线连接切换器件的功能来确保要使用的区域是连续的。这意味着,如附图标记 901 所示,与数据处理装置 (6) 和 (7) 对应的区域在被替换的同时在高速缓存存储器 900 上被确保。于是,即使当“ID”=“3”和“4”时,也可如“ID”=“1”和“2”的情况中那样通过简单的 FIFO 控制来获得对于高速缓存存储器 900 的数据访问。高速缓存存储器仲裁单元 180 获取根据读出“ID”的上述的基本指针和行数以及各 ID 的写入指针。当确定高速缓存命中时,高速缓存存储器仲裁单元 180 基于同时读出的结果“tag_id”以及高速缓存存储器 900 的基本指针和写入指针来计算地址。然后,高速缓存存储器仲裁单元 180 将读出的高速缓存数据作为数据“write_data”182 写入在由高速缓存存储器 900 的基本指针和写入指针的总值所指示的存储区域中。

[0138] 如上所述,高速缓存确定单元 500 使用流水线配置执行数据处理。通过计算与各自数据处理装置对应的输出的反转符号位和分割信息的至多 1 位逻辑积,可以确定高速缓存命中。结果,与常规的高速缓存装置相比,该高速缓存装置可以以非常高的操作频率来操作。由于从较旧的数据依次自动删除数据,因此不需要高速缓存冲突时的复杂替换控制。并

且,高速缓存存储器对于预定的区域仅需要环形FIFO操作。为了实现高速缓存存储器900,仅需要使用具有多个写入指针的多储蓄库(multi-bank)FIFO。并且,可使用多个FIFO来实现高速缓存存储器900。在以上描述的例子中,准备分割信息表,并且,对于各处理装置切换分割信息。但是,当各处理装置对于处理装置的各自处理内容(操作状态)指定分割信息时,可以实现以上的处理。

[0139] 图5中描述的多个数据处理装置(1)至(N)的第一流水线300和第二流水线360的各自节点基于一个系统的时钟信号(未示出)而操作。但是,可通过将不同系统的同相时钟信号分别输入到数据处理装置(1)至(N),操作多个数据处理装置(1)至(N)。在优先节电时,对于数据处理装置(1)至(N)中的一个的时钟信号被停止,以停止该数据处理装置。例如,在上述的实际例子800中,当只有处理装置A 101操作并且处理装置B 102至D 104停止时,在分割信息表515中的与“ID”=“2”至“4”对应的各条分割信息中设置信息“00000000”。然后,数据处理装置(1)以外的数据处理装置被停止。另一方面,为了在优先节电的同时操作处理装置A 101至D 104(尽管是以低速),在与“ID”=“1”至“4”对应的各条分割信息中设置信息“10000000”。通过该设置,数据处理装置(1)被所有的处理装置A 101至D 104共享。由于不使用剩余的数据处理装置,因此可以停止它们。并且,通过对于数据处理装置(1)至(N)的各自区域使用不同SRAM来实现高速缓存存储器900。对应于停止的数据处理装置,停止对于不使用的SRAM的时钟信号(或电源电压)的供给。以此方式,可以获得进一步的节电。

[0140] 图10是用于解释替代环形总线使用纵横开关的布置的示例性电路图。图10示出M个处理装置(1)至(M)和N个数据处理装置(1)至(N)之间的连接的例子,替代上述的经由环形总线的连接,经由纵横开关1020和1060获得所述连接。为了允许所述多个处理装置专用数据处理装置中的每一个,在数据处理装置之前和之后插入基于分割信息“partition”切换的纵横开关1020和1060。通过该布置,多个处理装置可并行使用多个数据处理装置。通过这样的并行处理,多个处理装置可被加速。

[0141] 也可通过读出并执行记录在存储器件上的程序以执行上述实施例的功能的系统或装置的计算机(或诸如CPU或MPU的器件)、以及通过由系统或装置的计算机通过例如读出并执行记录在存储器件上的程序以执行上述实施例的功能来执行其步骤的方法,实现本发明的各方面。出于该目的,例如经由网络或从用作存储器件的各种类型的记录介质(例如,计算机可读介质)给计算机提供程序。

[0142] 虽然已参照示例性实施例描述了本发明,但要理解,本发明不限于公开的示例性实施例。所附的权利要求的范围要被赋予最宽的解释,以包含所有这样的修改以及等同的结构和功能。

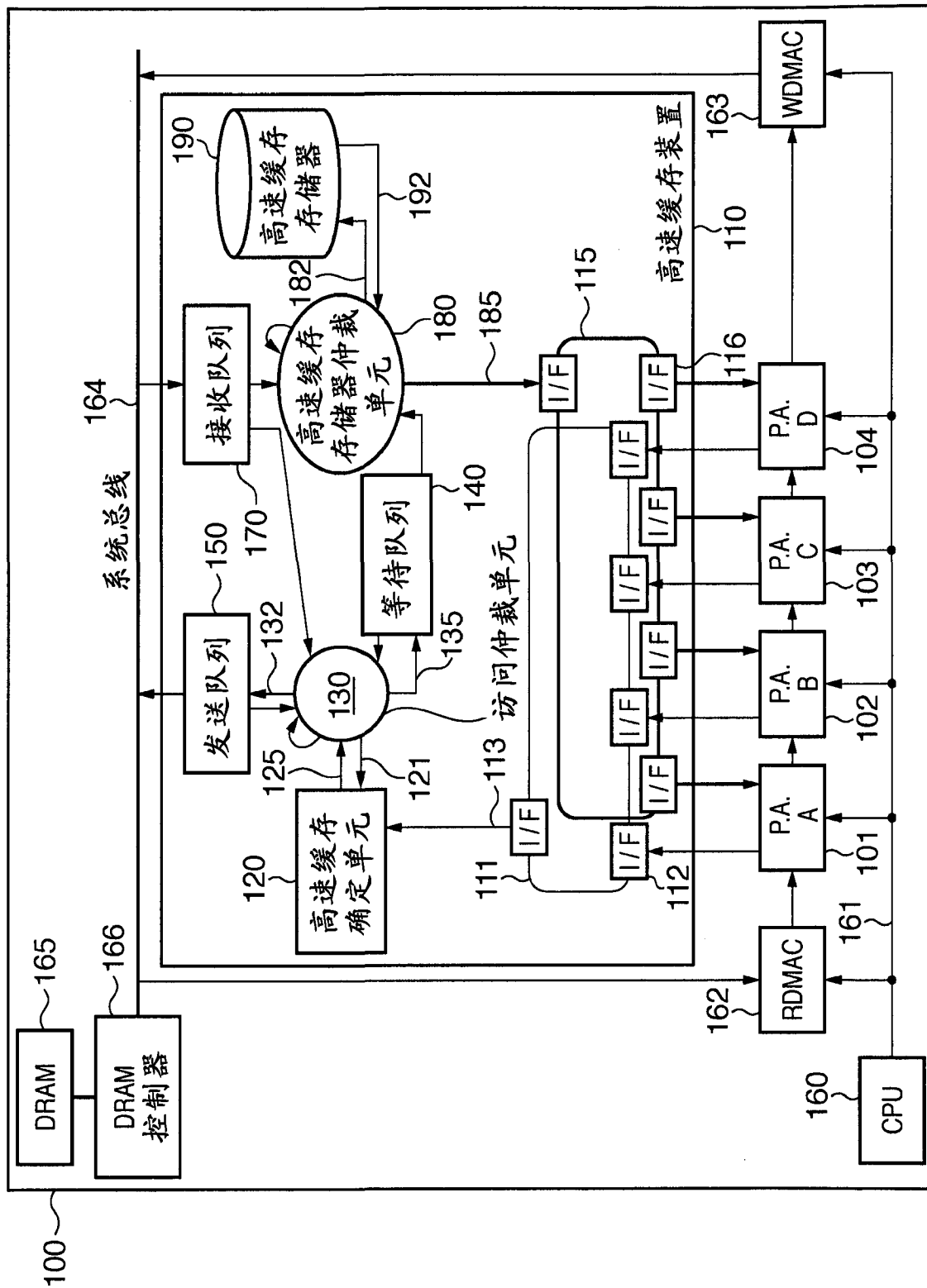


图 1

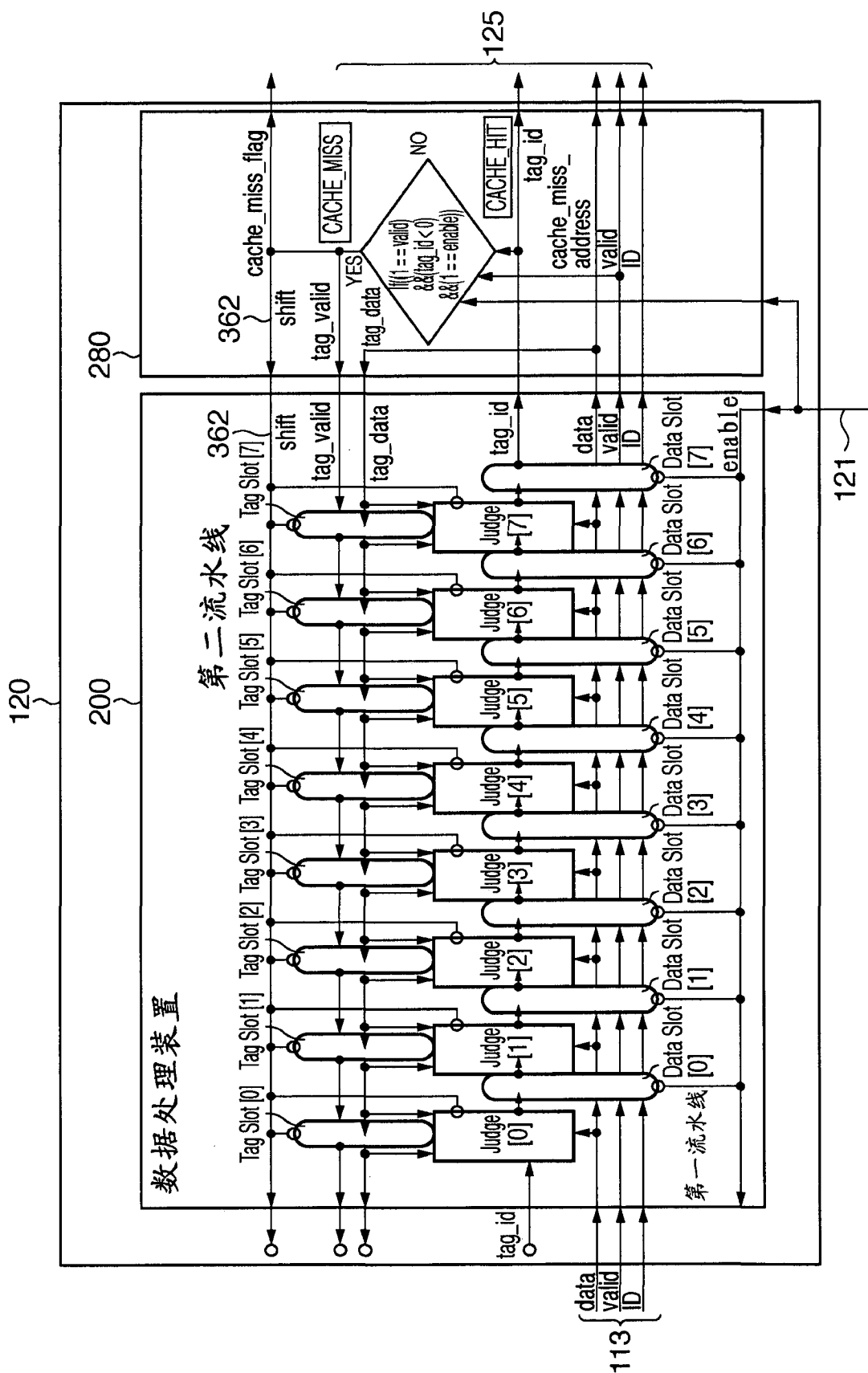


图 2

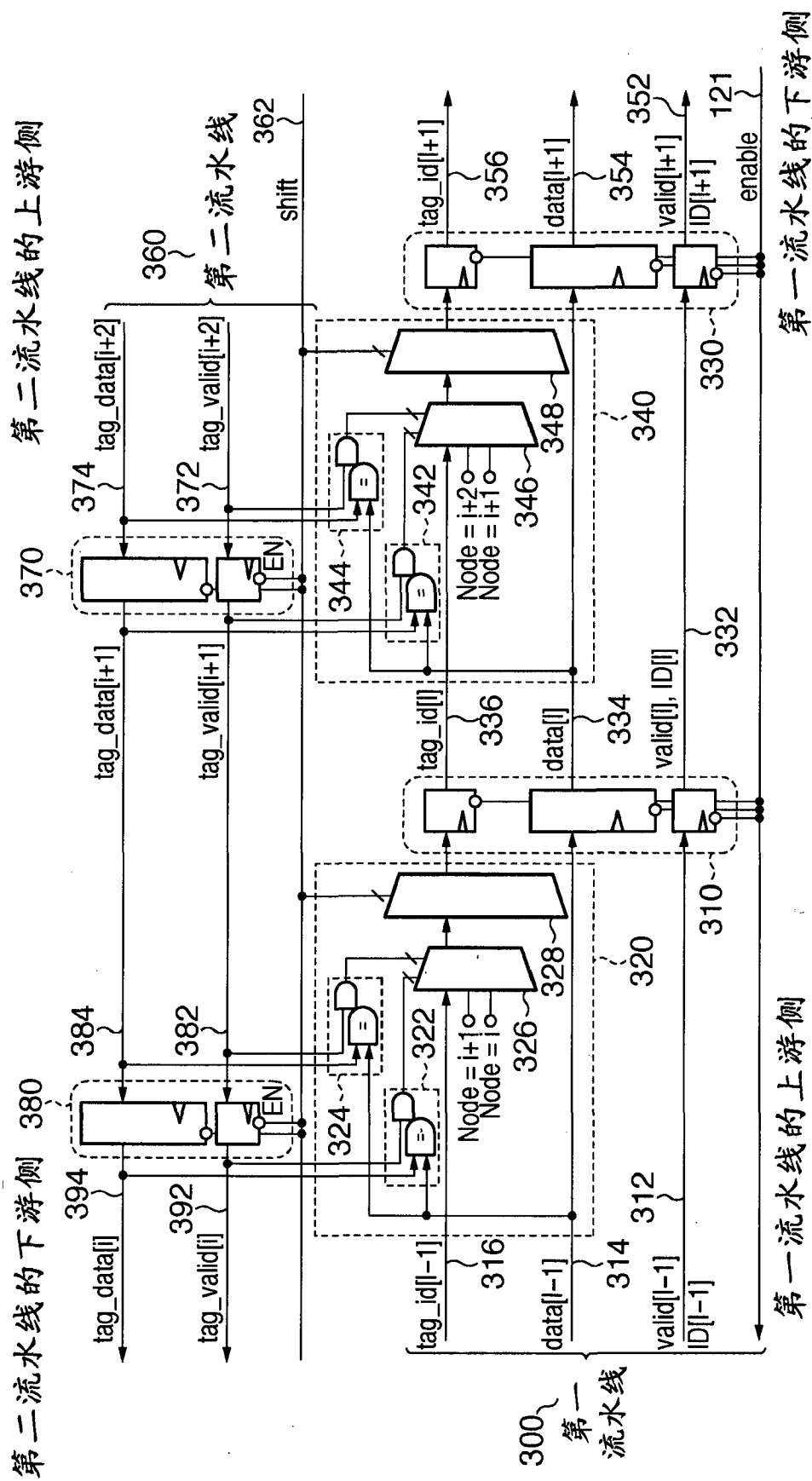


图 3

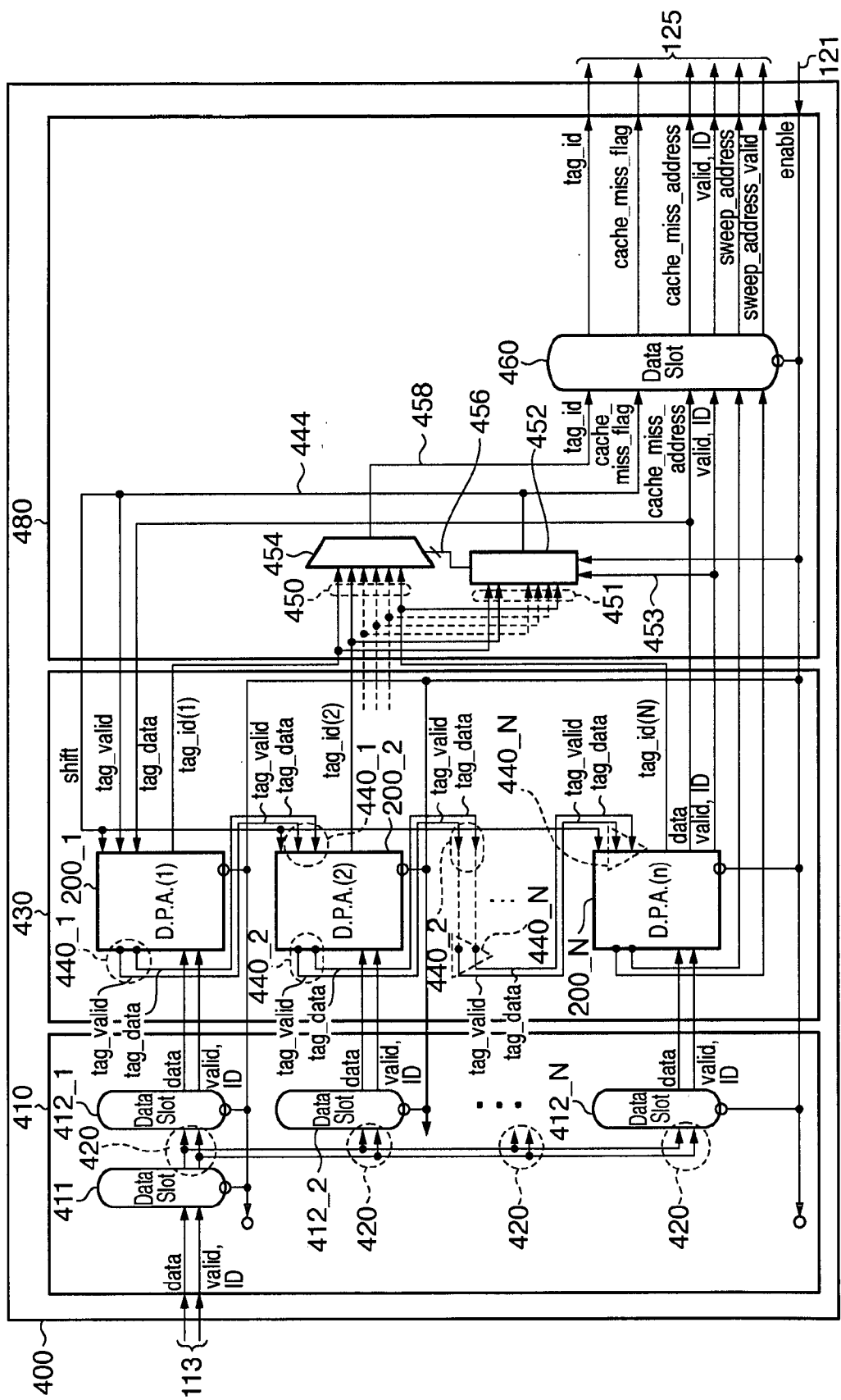


图 4

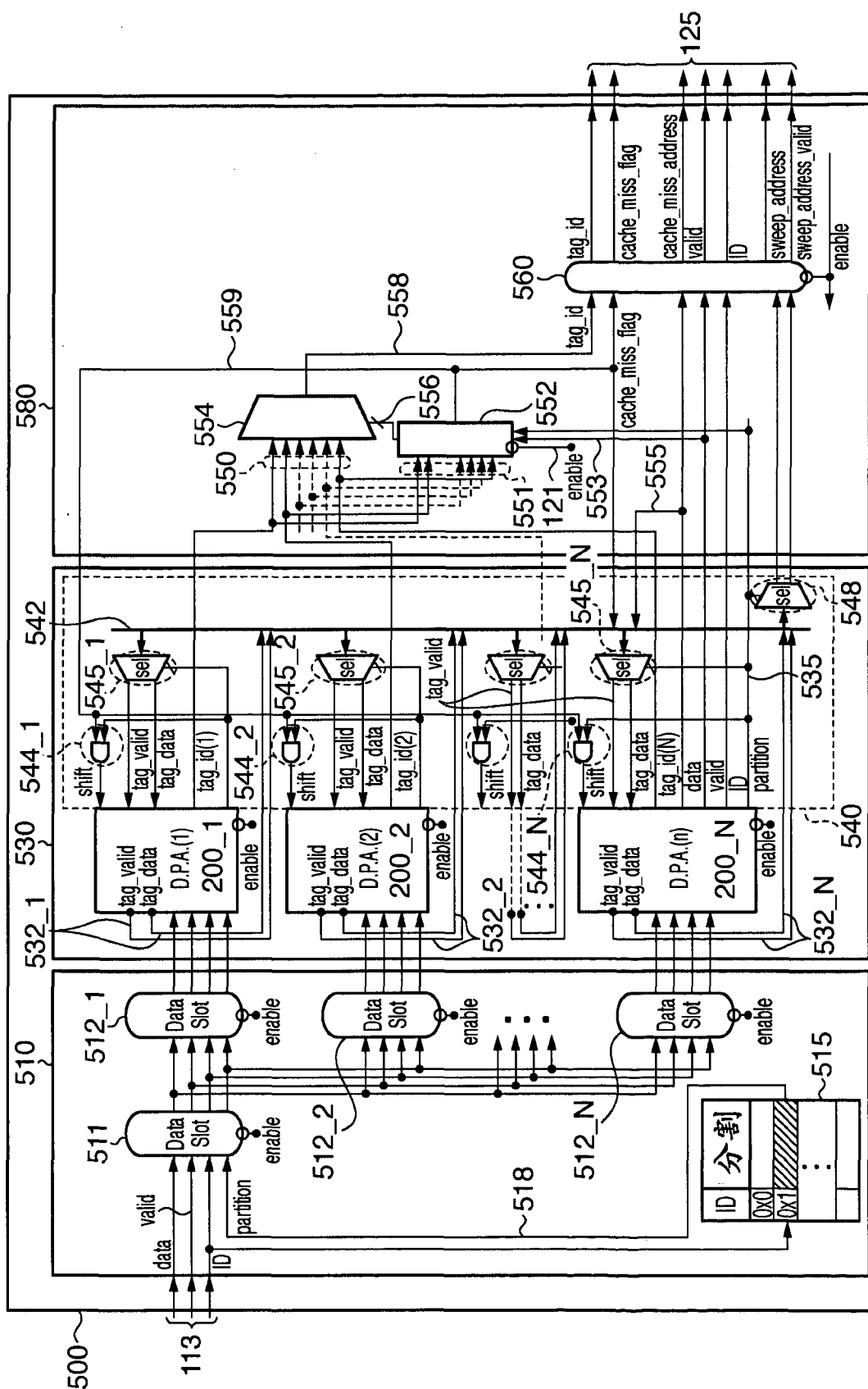


图 5

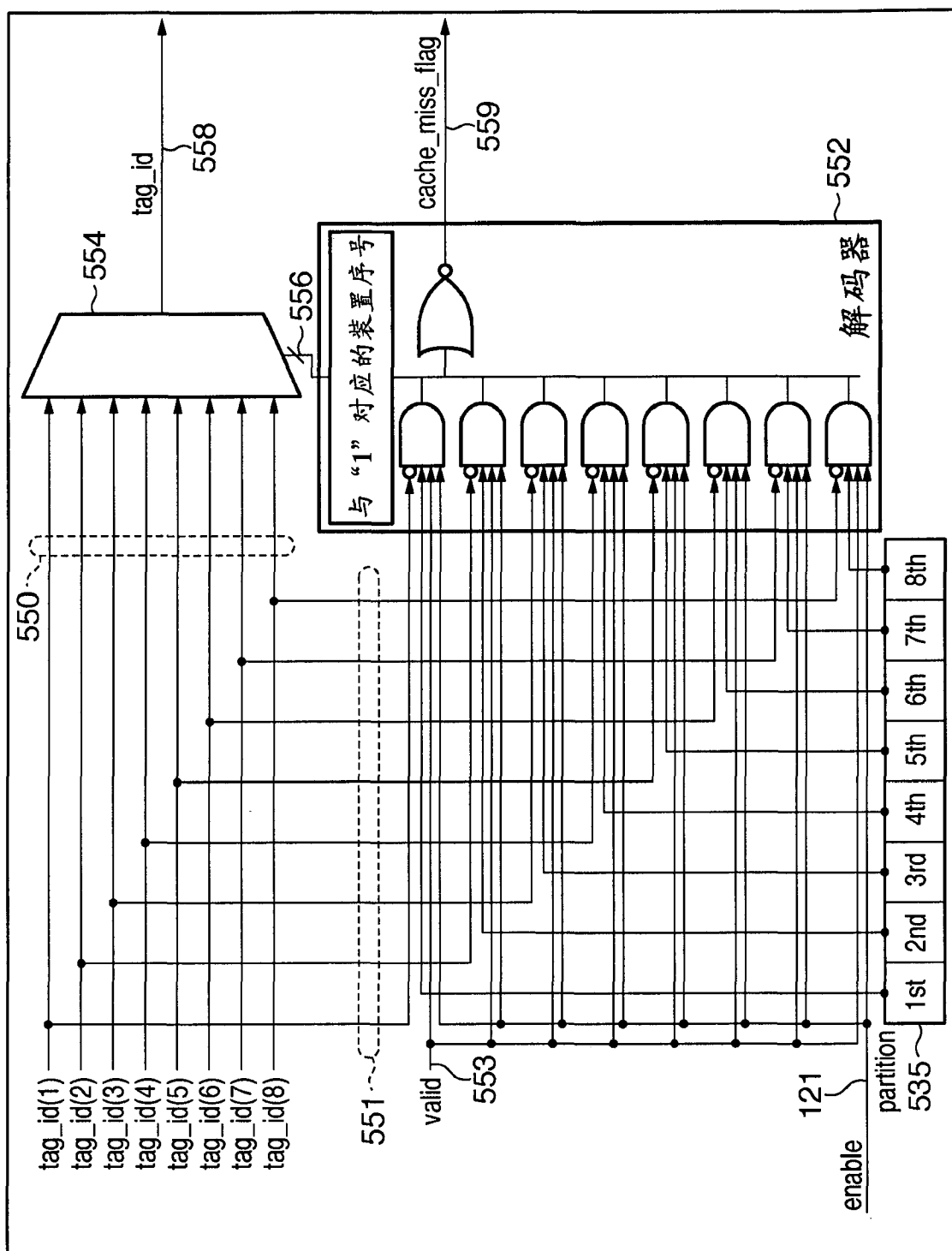


图 6

<格式>

700

702

704

处理装置识别信号 ID	分割信息[N 位]
1	110····100
2	000····001
3	001····010
⋮	⋮
M	000····000

图 7

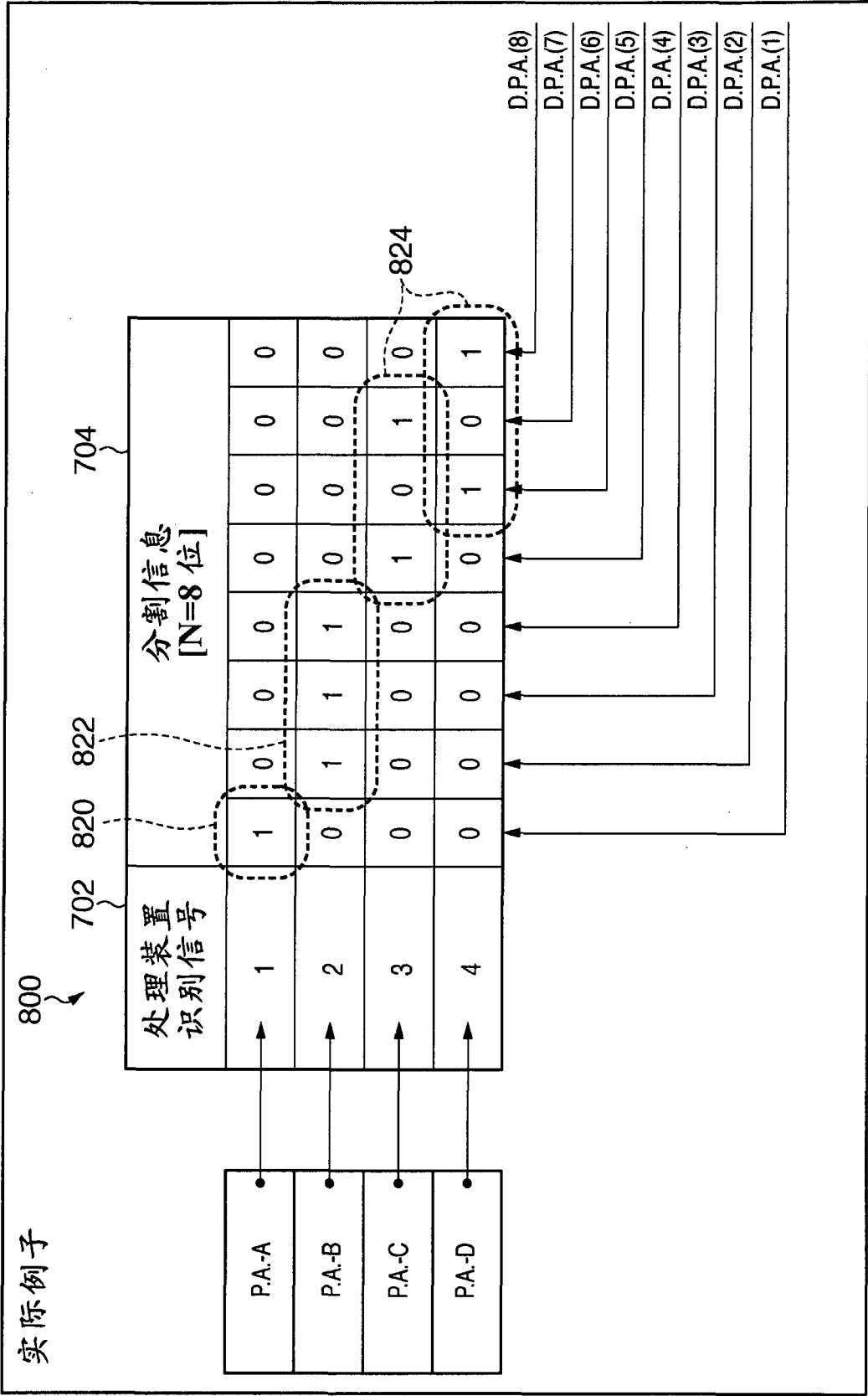


图 8

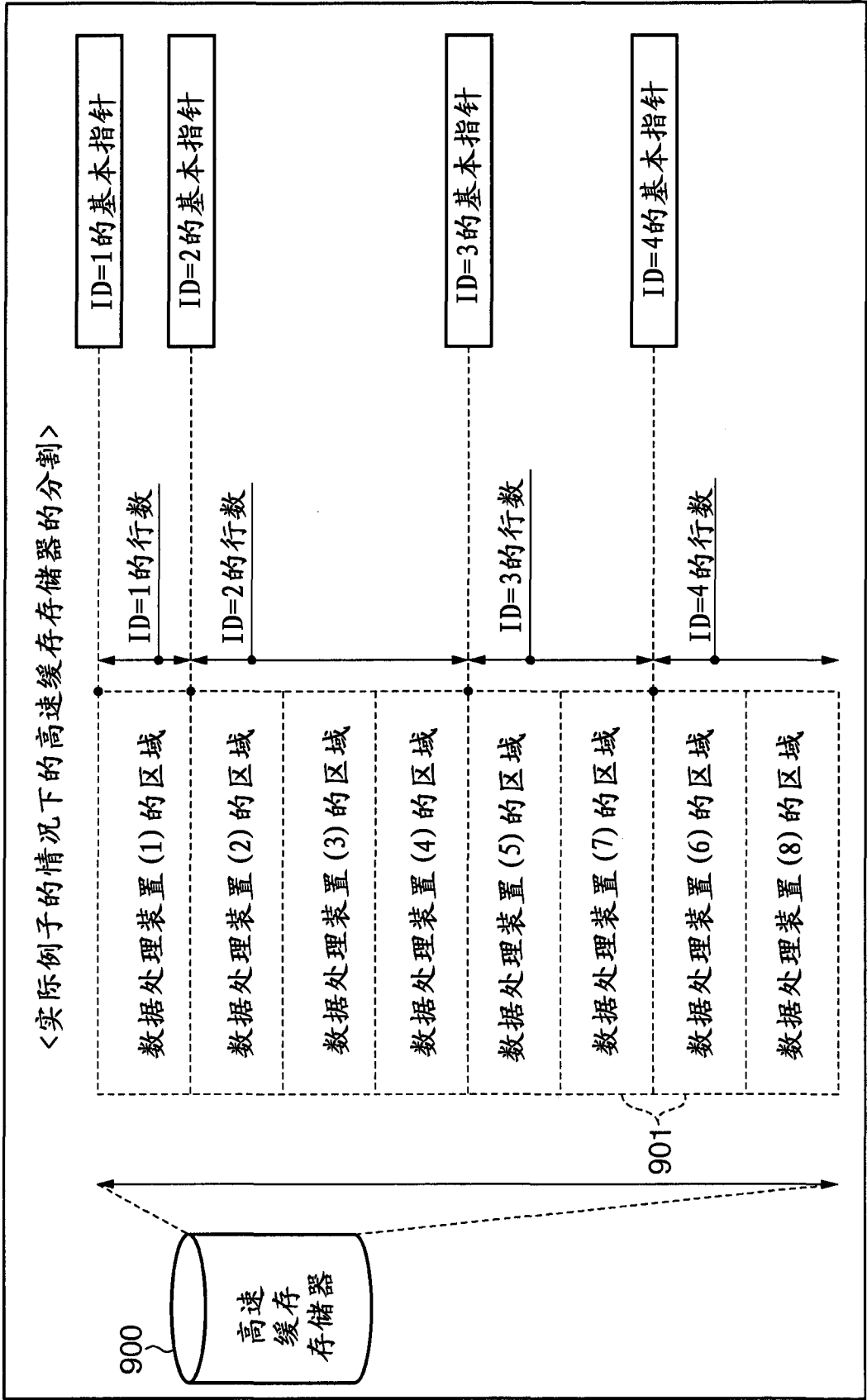


图 9

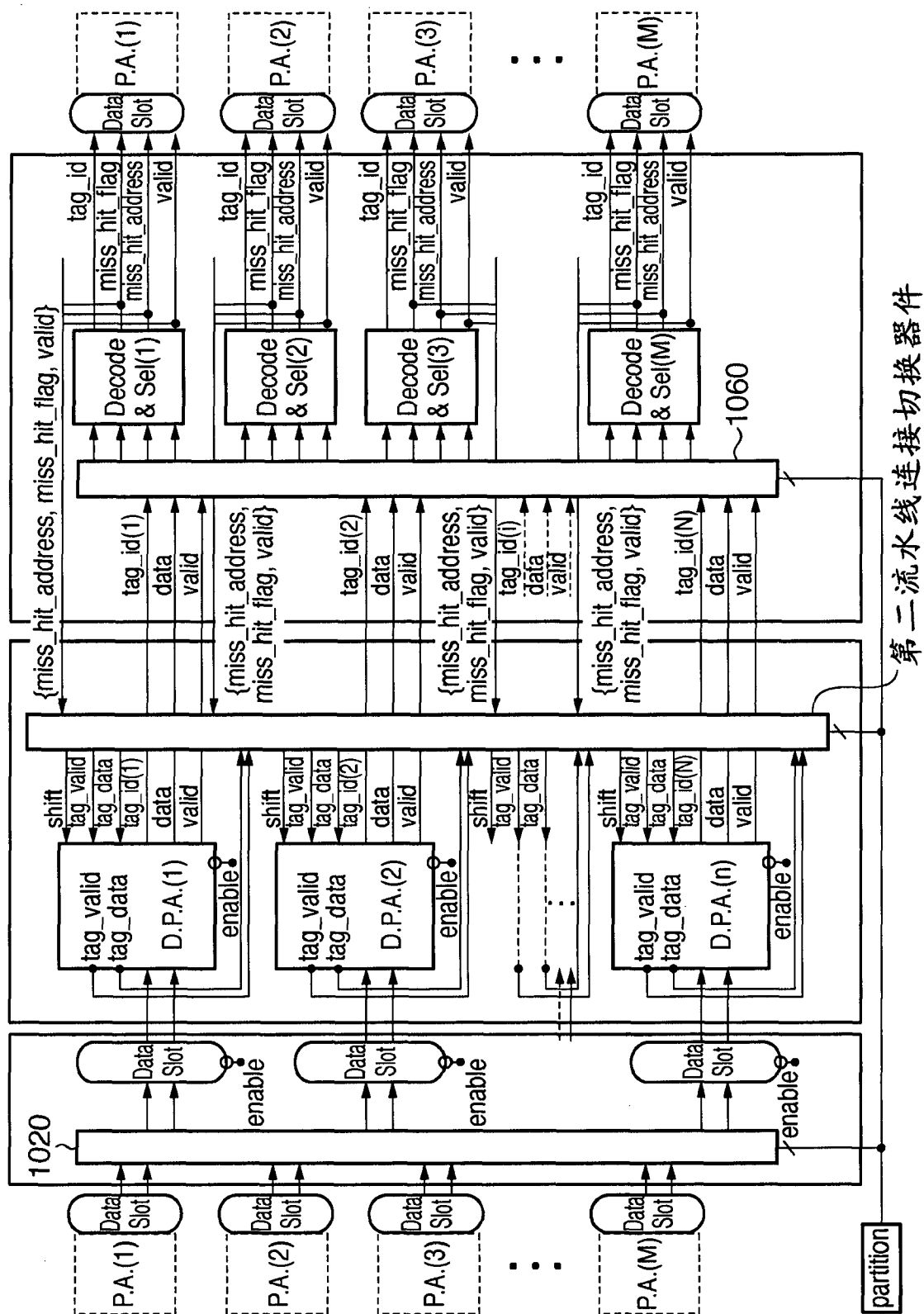


图 10

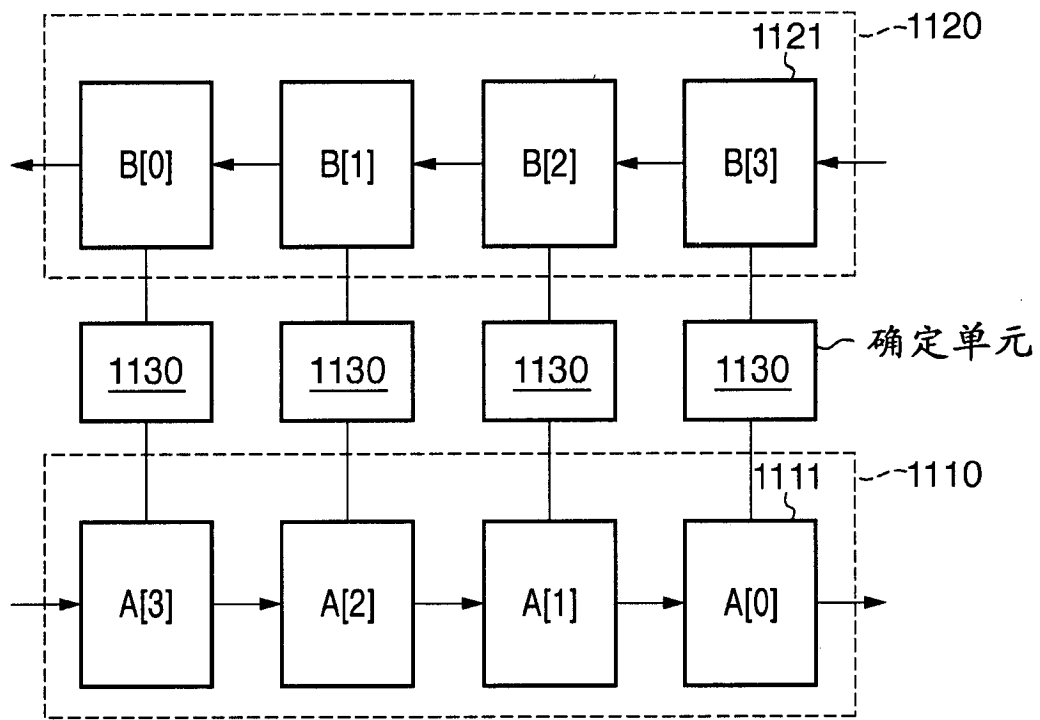


图 11