



(12) 发明专利

(10) 授权公告号 CN 103003792 B

(45) 授权公告日 2016. 05. 04

(21) 申请号 201180034987. 3

(51) Int. Cl.

(22) 申请日 2011. 06. 09

G06F 9/38(2006. 01)

(30) 优先权数据

61/353, 546 2010. 06. 10 US

12/943, 551 2010. 11. 10 US

(56) 对比文件

CN 101099147 B, 2010. 05. 19,

US 7693853 B2, 2010. 04. 06,

US 5081608 A, 1992. 01. 14,

US 2010121643 A1, 2010. 05. 13,

(85) PCT国际申请进入国家阶段日

2013. 01. 16

审查员 王燕

(86) PCT国际申请的申请数据

PCT/US2011/039849 2011. 06. 09

(87) PCT国际申请的公布数据

W02011/156634 EN 2011. 12. 15

(73) 专利权人 美光科技公司

地址 美国爱达荷州

(72) 发明人 保罗·德卢戈施

(74) 专利代理机构 北京律盟知识产权代理有限

责任公司 11287

代理人 宋献涛

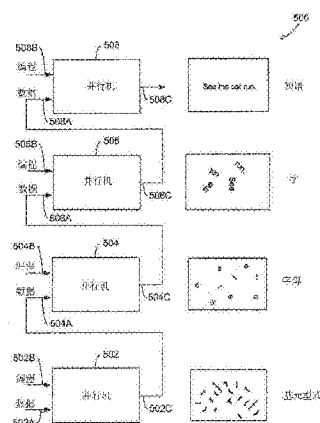
权利要求书5页 说明书16页 附图13页

(54) 发明名称

使用层次结构分析数据

(57) 摘要

本发明描述用于分析数据的设备、系统及方法。可使用层次结构来分析所述数据。一个此种层次结构可包含多个层,其中每一层对输入数据执行分析并基于所述分析而提供输出。可将来自所述层次结构中的较低层的输出作为输入提供到较高层。以此方式,较低层可执行较低层级的分析(例如,较基本/基础分析),而较高层可使用来自一个或一个以上较低层的所述输出执行较高层级的分析(例如,较复杂分析)。在一实例中,所述层次结构执行型式辨识。



1. 一种用于分析数据的方法,其包含:

通过确定第一数据是否匹配第一型式,使用在多个硬件元件上执行的第一并行机分析第一数据;

从所述第一并行机输出第二数据,所述第二数据从对所述第一数据的分析中产生并提供分析所述第一数据的结果的指示;

通过确定所述第二数据是否匹配第二型式,使用在所述多个硬件元件上执行的第二并行机分析所述第二数据;及

从所述第二并行机输出分析所述第二数据的结果;以及

将来自所述第二并行机的输出反馈到所述第一并行机,且基于经由所述第二并行机在所述第二数据中辨识出的所述第二型式来配置所述第一并行机用于后续操作。

2. 根据权利要求1所述的方法,其中所述第二数据提供所述第一数据是否匹配所述第一型式的指示。

3. 根据权利要求1所述的方法,其中分析所述第二数据的所述结果包含所述第二数据是否匹配所述第二型式的指示。

4. 根据权利要求1所述的方法,其中所述第二数据包含指示所述第一型式的匹配的单个位。

5. 根据权利要求1所述的方法,其中所述第二数据包含字,所述字包含指示与多个型式的匹配及不匹配的多个位,其中所述多个型式包括所述第一型式。

6. 根据权利要求1所述的方法,其中所述第二数据包含对应于所述第一并行机的可编程元件的状态的状态向量。

7. 根据权利要求1所述的方法,其中分析第一数据包含:确定所述第一数据是否包括所关注序列。

8. 根据权利要求1所述的方法,其中从所述第一并行机输出第二数据包含:基于所述第一并行机的可编程元件对所述第一数据的反应而输出第二数据。

9. 根据权利要求1所述的方法,其中分析第一数据包含:确定是否在所述第一数据中辨识出数据序列。

10. 根据权利要求1所述的方法,其中所述第一数据包含原始数据。

11. 根据权利要求1所述的方法,其中分析所述第二数据还包括:使用所述第二并行机分析所述第一数据。

12. 根据权利要求1所述的方法,其中所述第二数据提供所述第一并行机的检测状态的指示。

13. 根据权利要求1所述的方法,其进一步包含:将分析所述第二数据的所述结果反馈到所述第二并行机,其中使用所述第二并行机分析所述第二数据进一步包含:对分析所述第二数据的所述结果进行分析。

14. 根据权利要求3所述的方法,其中所述第二型式包含在所述第一数据中辨识出的型式中的一型式。

15. 根据权利要求1所述的方法,其中配置所述第一并行机包含:

分析在所述第二数据中辨识出的所述第二型式,且基于在所述第二数据中辨识出的所述第二型式而编译用于所述第一并行机的程序。

16. 根据权利要求 15 所述的方法,其中所述程序包含经更新程序。

17. 一种用于分析数据的方法,包括:

通过确定第一数据是否匹配第一型式,使用在多个硬件元件上执行的第一并行机分析第一数据;

从所述第一并行机输出第二数据,所述第二数据提供分析所述第一数据的结果的指示;

通过确定所述第二数据是否匹配第二型式,使用在所述多个硬件元件上执行的第二并行机分析所述第二数据;及

确定所述第二数据是否匹配第二多个型式中的一者或一者以上;

从所述第二并行机输出第三数据,所述第三数据提供分析所述第二数据的结果的指示,所述第三数据提供是否在所述第二数据中辨识出所述第二多个型式中的一者或一者以上的指示;

将来自所述第二并行机的所述第三数据反馈到所述第一并行机,且基于经由所述第二并行机在所述第二数据中辨识出的所述第二型式来配置所述第一并行机用于后续操作;

使用第三并行机确定所述第三数据是否匹配第三多个型式中的一者或一者以上;及

从所述第三并行机输出是否在所述第三数据中辨识出所述第三多个型式中的一者或一者以上的指示。

18. 根据权利要求 17 所述的方法,其进一步包含:

基于来自所述第三并行机的输出而配置所述第一并行机及所述第二并行机中的至少一者。

19. 一种用于处理数据的设备,其包含:

第一并行机,其经配置以在多个硬件元件上执行,所述第一并行机具有第一数据输入端口及第一输出端口,且所述第一并行机经配置以:

在所述第一数据输入端口处接收第一数据;及

在所述第一输出端口处输出第二数据,所述第二数据提供是否在所述第一数据中辨识出第一型式的指示;及

第二并行机,其经配置以在所述多个硬件元件上执行,所述第二并行机具有第二数据输入端口及第二输出端口,其中所述第二数据输入端口耦合到所述第一并行机的所述第一输出端口,所述第二并行机经配置以:

在所述第二数据输入端口处从所述第一并行机接收所述第二数据;及

在所述第二输出端口处输出第三数据,所述第三数据提供是否在所述第二数据中辨识出第二型式的指示;

其中将来自所述第二并行机的所述第三数据反馈到所述第一并行机,且基于经由所述第二并行机在所述第二数据中辨识出的所述第二型式来配置所述第一并行机用于后续操作;且

其中所述第一并行机包括经配置以基于所述第三数据而接收程序的编程接口,所述程序经配置以重新编程所述第一并行机。

20. 根据权利要求 19 所述的设备,其进一步包含:

第三并行机,其具有第三数据输入端口及第三输出端口,其中所述第三输出端口耦合

到所述第二并行机的所述第二数据输入端口。

21. 根据权利要求 19 所述的设备,其中所述第二并行机进一步经配置以

在所述第二数据输入端口处接收第一数据,且进一步经配置以在所述第二数据与所述第一数据的组合中辨识所述第二型式。

22. 根据权利要求 19 所述的设备,其中所述程序包含经更新程序,其中所述经更新程序经配置以更新已加载到所述第一并行机上的程序。

23. 根据权利要求 19 所述的设备,其中所述程序包含用于所述第一并行机的原始程序。

24. 根据权利要求 19 所述的设备,其进一步包含:

处理装置,其耦合到所述第二并行机的所述输出端口且经配置以分析所述第二并行机的所述输出端口并依据从所述第二并行机输出的第二数据而编译用于所述第一并行机的更新程序;

其中所述第一并行机经配置以从所述处理装置接收经更新程序。

25. 根据权利要求 24 所述的设备,其包含:

第三并行机,其具有第三数据输入端口及第三输出端口,所述第三并行机经配置以:
从所述第二并行机接收所述指示;
确定第三型式是否在所述指示中;及
产生输出。

26. 一种用于分析数据的方法,其包含:

在多个硬件元件上执行的第一并行机处接收第一数据;
从所述第一并行机输出第二数据,所述第二数据从对所述第一数据的分析中产生;
在所述多个硬件元件上执行的第二并行机处接收所述第二数据;
从所述第二并行机输出第三数据,所述第三数据从对所述第二数据的分析中产生;及
将来自所述第二并行机的所述第三数据反馈到所述第一并行机,且基于所述第三数据而配置所述第一并行机,所述第三数据用于所述第一并行机随后的操作;

其中所述第一并行机经配置以在所述第一数据中识别第一型式,且其中所述第二并行机经配置以在所述第二数据中识别第二型式。

27. 根据权利要求 26 所述的方法,其中配置包含重新编程。

28. 根据权利要求 26 所述的方法,其中配置所述第一并行机包含:

基于所述第三数据而编译用于所述第一并行机的经更新程序;及
将所述更新程序加载到所述第一并行机上。

29. 根据权利要求 26 所述的方法,其进一步包含:

在所述第二并行机处接收所述第一数据,其中从所述第二并行机输出第三数据包含:
输出第三数据,所述第三数据从对所述第一和所述第二数据的分析中产生。

30. 一种用于分析数据的方法,其包含:

使用在多个硬件元件上执行的层次结构的第一层中的第一并行机,对输入数据执行较低层级的型式辨识分析;

基于所述较低层级的型式辨识分析而由第一层提供第一输出;

使用在所述多个硬件元件上执行的层次结构的第二层中的第二并行机,对第一输出执

行较高层级的型式辨识分析；

基于对所述第一输出的所述较高层级的型式辨识分析而由第二层提供第二输出；以及将反馈信息从所述层次结构的所述第二层提供到所述层次结构的所述第一层，且基于所述第一输出或者第二输出分别重新配置所述第一层或者所述第二层中的至少一者，来修改对所述第一层或者所述第二层的随后的型式辨识分析，其中所述反馈信息基于所述第二输出而生成。

31. 根据权利要求 30 所述的方法，其中以级联方式将所述第一与第二并行机耦合在一起。

32. 根据权利要求 30 所述的方法，其中将所述第一与第二并行机串联耦合。

33. 根据权利要求 30 所述的方法，其中使用所述反馈信息来更新所述较低层级的型式辨识分析。

34. 根据权利要求 33 所述的方法，其中使用所述反馈信息来更新用以执行所述较低层级的型式辨识分析的程序。

35. 根据权利要求 33 所述的方法，其中所述反馈信息包含对应于所述输入数据的识别信息。

36. 根据权利要求 30 所述的方法，其中所述层次结构包含第三并行机。

37. 根据权利要求 30 所述的方法，其进一步包含：基于所述较高层级的型式辨识分析而配置所述较低层级的型式辨识分析。

38. 根据权利要求 30 所述的方法，其进一步包含：基于所述较低层级的型式辨识分析而配置所述较高层级的型式辨识分析。

39. 根据权利要求 30 所述的方法，其中在具有冯·诺伊曼架构的机器上以软件来实施所述第一层和第二层。

40. 根据权利要求 30 所述的方法，其中所述第一输出包含检测状态信息。

41. 一种用于处理数据的设备，其包含：

第一并行机，其经配置以在多个硬件元件上执行，所述第一并行机具有第一数据输入端口及第一输出端口，且所述第一并行机经配置以：

在所述第一数据输入端口处接收第一数据；及

基于是否在所述第一数据中辨识出第一型式而在所述第一输出端口处输出第二数据；及

第二并行机，其经配置以在多个硬件元件上执行，所述第二并行机具有第二数据输入端口及第二输出端口，其中所述第二数据输入端口耦合到所述第一并行机的所述第一输出端口，所述第二并行机经配置以：

在所述第二数据输入端口处接收所述第二数据；及

在所述第二输出端口处输出是否在所述第二数据中辨识出第二型式的指示；

处理装置，其耦合到所述第二并行机的所述输出端口且经配置以分析所述第二并行机的所述输出端口并依据从所述第二并行机输出的第二数据而编译用于所述第一并行机的更新程序；

其中将来自所述第二并行机的输出反馈到所述第一并行机，且基于经由所述第二并行机在所述第二数据中辨识出的所述第二型式来配置所述第一并行机用于后续操作；且

其中所述第一并行机经配置以从所述处理装置接收经更新程序。

42. 根据权利要求 41 所述的设备,其包含:

第三并行机,其经配置以在多个硬件元件上执行,所述第三并行机具有第三数据输入端口及第三输出端口,所述第三并行机经配置以:

从所述第二并行机接收所述指示;

确定第三型式是否在所述指示中;及

产生输出。

使用层次结构分析数据

[0001] 相关申请案的交叉参考

[0002] 本专利申请案主张 2010 年 11 月 10 日提出申请的标题为“使用层次结构分析数据 (Analyzing Data Using A Hierarchical Structure)”的第 12/943,551 号美国申请案的优先权权益,所述美国申请案依据 35U. S. C. 第 119(e) 款主张 2010 年 6 月 10 日提出申请的标题为“层次型式辨识 (Hierarchical Pattern Recognition)”的第 61/353,546 号美国临时专利申请案的优先权权益,所述申请案两者均以全文引用的方式特此并入本文中。

技术领域

背景技术

[0003] 复杂型式辨识在常规的基于冯·诺依曼的计算机上执行起来可能效率低下。然而,生物大脑,特定来说人类大脑擅长执行型式辨识。当前研究表明,人类大脑使用大脑新皮质中的一系列以层次方式组织的神经元层来执行型式辨识。层次的较低层中的神经元分析来自(例如)感觉器官的“原始信号”,而较高层中的神经元分析来自较低层级中的神经元的信号输出。大脑新皮质中的此层次系统可能与大脑的其它区域组合来实现复杂型式辨识,此使得人类能够执行高级功能,例如空间推理、意识思考及复杂语言。

发明内容

附图说明

- [0004] 图 1 图解说明根据本发明的各种实施例的并行机的实例。
- [0005] 图 2 图解说明根据本发明的各种实施例的有限状态机的实例。
- [0006] 图 3 图解说明根据本发明的各种实施例的借助并行机所实施的两层级层次的实例。
- [0007] 图 4 图解说明根据本发明的各种实施例的借助并行机所实施的两层级层次的另一实例。
- [0008] 图 5 图解说明根据本发明的各种实施例的借助并行机所实施的四层级层次的实例。
- [0009] 图 6 图解说明根据本发明的各种实施例的借助并行机所实施的具有反馈的四层级层次的实例。
- [0010] 图 7 图解说明根据本发明的各种实施例的借助并行机所实施的具有反馈的四层级层次的另一实例。
- [0011] 图 8 图解说明根据本发明的各种实施例的实施为有限状态机引擎的图 1 的并行机的实例。
- [0012] 图 9 图解说明根据本发明的各种实施例的图 8 的有限状态机引擎的块的实例。
- [0013] 图 10 图解说明根据本发明的各种实施例的图 9 的块的一行的实例。

[0014] 图 11 图解说明根据本发明的各种实施例的含图 10 的行中的两个一群组的实例。

[0015] 图 12 图解说明根据本发明的各种实施例的用于编译器将源代码转换成图像以用于编程图 8 的并行机的方法的实例。

[0016] 图 13 图解说明根据本发明的各种实施例的具有基于冯·诺伊曼的架构的计算机的实例。

具体实施方式

[0017] 以下描述及图式充分图解说明特定实施例以使所属领域的技术人员能够实践所述特定实施例。其它实施例可并入有结构、逻辑、电、过程及其它改变。一些实施例的各部分及特征可包括于其它实施例的那些部分及特征中或替代其它实施例的那些部分及特征。

[0018] 本文件除其它之外还描述用于使用层次结构分析数据的方法及设备。所述层次结构可包含多个层，其中每一层对输入数据执行分析并基于所述分析而提供输出。可将来自所述层次结构中的较低层的输出作为输入提供到较高层。以此方式，较低层可执行较低层级的分析（例如，较基本 / 基础分析），而较高层可使用来自一个或一个以上较低层的输出执行较高层级的分析（例如，较复杂分析）。在一实例中，所述层次结构执行型式辨识。在一实例中，型式辨识包括识别符号序列。用于型式的识别的实例性符号可对应于音素（音频）、图像中的像素、ASCII 字符、机器数据（例如，0 及 1）。

[0019] 在一实例中，借助以级联方式耦合在一起的多个并行机来实施所述层次结构。举例来说，第一并行机与第二并行机可串行耦合，以使得所述第二并行机接收来自所述第一并行机的输出作为输入。任一数目个并行机可一起耦合在此层次结构中。

[0020] 除了使用层次结构分析数据之外，本文件还描述用于使用来自在层次的一个层级处执行的分析的信息来修改在所述层次的另一层级处执行的分析的方法及设备。使用上文所述的并行机实例，实施较高层级的分析的第二并行机可将反馈信息提供到实施较低层级的分析的第一并行机。所述反馈信息可由第一并行机用来以类似于在生物大脑中学习的方式更新第一并行机所执行的分析。

[0021] 图 1 图解说明可用以实施用于分析数据的层次结构的实例性并行机 100。并行机 100 可接收输入数据并基于所述输入数据而提供输出。并行机 100 可包括用于接收输入数据的数据输入端口 110 及用于将输出提供到另一装置的输出端口 114。数据输入端口 110 提供用于将数据输入到并行机 100 的接口。

[0022] 并行机 100 包括多个可编程元件 102，每一可编程元件 102 具有一个或一个以上输入 104 及一个或一个以上输出 106。可编程元件 102 可被编程到多个状态中的一者中。可编程元件 102 的状态确定可编程元件 102 将基于给定输入而提供何种输出。也就是说，可编程元件 102 的状态确定可编程元件将基于给定输入而如何做出反应。可将输入到数据输入端口 110 的数据提供到多个可编程元件 102 以致使可编程元件 102 对其采取行动。可编程元件 102 的实例可包括下文详细论述的状态机元件（SME）及可配置逻辑块。在一实例中，SME 可被设定成给定状态以当在数据输入端口 110 处接收到给定输入时提供某一输出（例如，高信号或“1”信号）。当在数据输入端口 110 处接收到除所述给定输入之外的输入时，所述 SME 可提供不同输出（例如，低信号或“0”信号）。在一实例中，可配置逻辑块可经设定以基于在数据输入端口 110 处所接收的输入而执行布尔逻辑函数（例如，“与”、“或”、“非

或”等)。

[0023] 并行机 100 还可包括用于将程序(例如,图像)加载到并行机 100 上的编程接口 111。所述图像可编程(例如,设定)可编程元件 102 的状态。也就是说,所述图像可配置可编程元件 102 以便以某一方式对给定输入做出反应。举例来说,可编程元件 102 可经设定以当在数据输入端口 110 处接收到字符‘a’时输出高信号。在一些实例中,并行机 100 可使用时钟信号来控制可编程元件 102 的操作的计时。在某些实施例中,并行机 100 可包括专用元件 112(例如,RAM、逻辑门、计数器、查找表等)以用于与可编程元件 102 交互且用于执行专用功能。在一些实施例中,在数据输入端口 110 处所接收的数据可包括随时间或同时一起接收的固定数据集或随时间接收的数据流。所述数据可从耦合到并行机 100 的任一源(例如数据库、传感器、网络等)接收或由所述源产生。

[0024] 并行机 100 还包括多个可编程开关 108 以用于选择性地 将并行机 100 的不同元件(例如,可编程元件 102、数据输入端口 110、输出端口 114、编程接口 111 及专用元件 112)耦合在一起。因此,并行机 100 包含在所述元件当中形成的可编程矩阵。在一实例中,可编程开关 108 可选择性地将两个或两个以上元件彼此耦合在一起以使得可编程元件 102 的输入 104、数据输入端口 110、编程接口 111 或专用元件 112 可通过一个或一个以上可编程开关 108 耦合到可编程元件 102 的输出 106、输出端口 114、编程接口 111 或专用元件 112。因此,可通过设定可编程开关 108 来控制信号在所述元件之间的路由。虽然图 1 图解说明给定元件与可编程开关 108 之间的某一数目个导体(例如,导线),但应理解,在其它实例中可使用不同数目个导体。此外,虽然图 1 图解说明每一可编程元件 102 个别地耦合到可编程开关 108,但在其它实例中,多个可编程元件 102 可作为一群组(例如,图 8 中所图解说明的块 802)耦合到可编程开关 108。在一实例中,数据输入端口 110、数据输出端口 114 及/或编程接口 111 可实施为寄存器以使得向所述寄存器的写入将数据提供到相应元件或从所述相应元件提供数据。

[0025] 在一实例中,在一物理装置上实施单个并行机 100,然而在其它实例中可在单个物理装置(例如,物理芯片)上实施两个或两个以上并行机 100。在一实例中,多个并行机 100 中的每一者可包括相异数据输入端口 110、相异输出端口 114、相异编程接口 111 及一组相异可编程元件 102。此外,每一组可编程元件 102 可对其对应输入数据端口 110 处的数据做出反应(例如,输出高信号或低信号)。举例来说,对应于第一并行机 100 的第一组可编程元件 102 可对对应于第一并行机 100 的第一数据输入端口 110 处的数据做出反应。对应于第二并行机 100 的第二组可编程元件 102 可对对应于第二并行机 100 的第二数据输入端口 110 做出反应。因此,每一并行机 100 包括一组可编程元件 102,其中不同组可编程元件 102 可对不同输入数据做出反应。类似地,每一并行机 100 及每一对应组可编程元件 102 可提供相异输出。在一些实例中,来自第一并行机 100 的输出端口 114 可耦合到第二并行机 100 的输入端口 110,以使得用于第二并行机 100 的输入数据可包括来自第一并行机 100 的输出数据。

[0026] 在一实例中,用于加载到并行机 100 上的图像包含用于设定可编程元件 102 的状态、编程可编程开关 108 及配置并行机 100 内的专用元件 112 的多个信息位。在一实例中,可将所述图像加载到并行机 100 上以编程并行机 100 以基于某些输入而提供所要输出。输出端口 114 可基于可编程元件 102 对数据输入端口 110 处的数据的反应而提供来自并行机

100 的输出。来自输出端口 114 的输出可包括指示给定型式的匹配的单个位、包含指示与多个型式的匹配及不匹配的多个位的字及对应于所有或某些可编程元件 102 在给定时刻的状态的状态向量。

[0027] 并行机 100 的实例性用途包括型式辨识（例如，语音辨识、图像辨识等）、信号处理、成像、计算机视觉、密码编译及其它。在某些实例中，并行机 100 可包含有限状态机 (FSM) 引擎、现场可编程门阵列 (FPGA) 及其变化形式。此外，并行机 100 可为较大装置（例如，计算机、寻呼机、蜂窝式电话、个人记事本、便携式音频播放器、网络装置（例如，路由器、防火墙、交换机或其任一组合）、控制电路、相机等）中的组件。

[0028] 图 2 图解说明可由并行机 100 实施的有限状态机 (FSM) 的实例性模型。并行机 100 可配置（例如，编程）为 FSM 的物理实施方案。FSM 可表示为图 200（例如，有向图、无向图、伪图），其含有一个或一个以上根节点 202。除了根节点 202 之外，所述 FSM 可由通过一个或一个以上边缘 206 连接到根节点 202 及其它标准节点 204 的数个标准节点 204 及终端节点 208 构成。节点 202、204、208 对应于所述 FSM 中的状态。边缘 206 对应于所述状态之间的转变。

[0029] 所述节点（202、204、208）中的每一者可处于活动或不活动状态中。当处于不活动状态中时，节点（202、204、208）不对输入数据做出反应（例如，响应）。当处于活动状态中时，节点（202、204、208）可对输入数据做出反应。当输入数据匹配上游节点（202、204）与在所述节点下游的节点（204、208）之间的边缘 206 所指定的准则时，所述上游节点（202、204）可通过激活所述下游节点（204、208）来对所述输入数据做出反应。举例来说，当指定字符 ‘b’ 的第一节点 204 为活动的且接收到字符 ‘b’ 作为输入数据时，所述第一节点 204 将激活通过边缘 206 连接到所述第一节点 204 的第二节点 204。如本文所使用，“上游”是指一个或一个以上节点之间的关系，其中在一个或一个以上其它节点上游（或在循环的情况中，在其自身上游）的第一节点是指所述第一节点可激活所述一个或一个以上其它节点（或在循环的情况中，可激活其自身）的情形。类似地，“下游”是指如下关系，在所述关系中在一个或一个以上其它节点下游（在循环的情况中，在其自身下游）的第一节点可由所述一个或一个以上其它节点激活（或在循环的情况中，可由其自身激活）。因此，本文使用术语“上游”及“下游”来指一个或一个以上节点之间的关系，但此些术语不排除循环的使用或节点当中的其它非线性路径。

[0030] 在图 200 中，根节点 202 可首先被激活且当输入数据匹配来自根节点 202 的边缘 206 时根节点 202 可激活下游节点 204、208。当接收到输入数据时可以此方式激活整个图 200 中的节点 204、208。终端节点 208 对应于输入数据与所关注序列的匹配。因此，激活终端节点 208 指示已接收到所关注序列作为输入数据。在实施型式辨识功能的并行机 100 的背景中，到达终端节点 208 可指示已在输入数据中检测到特定所关注型式。

[0031] 在一实例中，每一根节点 202、标准节点 204 及终端节点 208 可对应于并行机 100 中的可编程元件 102。每一边缘 206 可对应于可编程元件 102 之间的连接。因此，转变到另一标准节点 204 或终端节点 208（例如，具有连接到另一标准节点 204 或终端节点 208 的边缘 206）的标准节点 204 对应于转变到另一可编程元件 102（例如，提供到另一可编程元件 102 的输出）的可编程元件 102。在一些实例中，根节点 202 不具有对应可编程元件 102。

[0032] 当将并行机 100 编程为 FSM 时，可编程元件 102 中的每一者还可处于活动或不活

动状态中。给定可编程元件 102 在不活动时不对其对应数据输入端口 110 处的输入数据做出反应。活动可编程元件 102 可对输入数据及数据输入端口 110 做出反应,且可在输入数据匹配下游可编程元件 102 的设定时激活所述可编程元件 102。当可编程元件 102 对应于终端节点 208 时,所述可编程元件 102 可耦合到输出端口 114 以将匹配的指示提供到外部装置。

[0033] 经由编程接口 111 加载到并行机 100 上的图像可配置可编程元件 102 及其它元件 112 以及可编程元件 102 与其它元件 112 之间的连接,以使得基于对数据输入端口 110 处的数据的反应而通过节点的顺序激活来实施所要 FSM。在一实例中,可编程元件 102 保持活动达单个数据循环(例如,单个字符、一组字符、单个时钟循环)且除非被上游可编程元件 102 重新激活否则接着切换到不活动。

[0034] 可认为终端节点 208 存储经压缩的过去事件历史。举例来说,到达终端节点 208 所需要的一个或一个以上输入数据类型可由所述终端节点 208 的激活表示。在一实例中,终端节点 208 所提供的输出为二进制的,也就是说,所述输出指示是否已匹配所关注型式。图 200 中终端节点 208 对标准节点 204 的比率可相当小。换句话说,虽然在所述 FSM 中可存在高复杂度,但比较来说所述 FSM 的输出可为小的。

[0035] 在一实例中,并行机 100 的输出可包含并行机的状态向量。所述状态向量包含并行机 100 的可编程元件 102 的状态(例如,被激活或未被激活)。在一实例中,所述状态向量包括对应于终端节点 208 的可编程元件 102 的状态。因此,输出可包括图 200 的所有终端节点 208 所提供的指示的集合。所述状态向量可表示为一字,其中每一终端节点 208 所提供的二进制指示包含所述字的一个位。终端节点 208 的此编码可提供并行机 100 的检测状态的有效指示(例如,是否已检测到所关注序列及已检测到何种所关注序列)。在另一实例中,状态向量可包括可编程元件 102 的全部或一子组的状态而不管所述可编程元件 102 是否对应于终端节点 208。

[0036] 如上文所提及,并行机 100 可经编程以实施型式辨识功能。举例来说,并行机 100 所实施的 FSM 可经配置以辨识输入数据中的一个或一个以上数据序列(例如,签名、型式)。当并行机 100 辨识出所关注数据序列时,可在输出端口 114 处提供所述辨识的指示。在一实例中,型式辨识可辨识一串符号(例如,ASCII 字符)以(例如)识别网络数据中的恶意软件或其它信息。

[0037] 图 3 图解说明经配置以使用层次结构 300 分析数据的第一并行机 302 及第二并行机 304 的实例。每一并行机 302、304 包括数据输入端口 302A、304A、编程接口 302B、304B 及输出端口 302C、304C。

[0038] 第一并行机 302 经配置以在数据输入端口 302A 处接收输入数据,例如原始数据。第一并行机 302 如上文所述对输入数据做出响应且在输出端口 302C 处提供输出。来自第一并行机 302 的输出发送到第二并行机 304 的数据输入端口 304A。第二并行机 304 接着可基于第一并行机 302 所提供的输出而做出反应且在输出端口 304C 处提供对应输出。两个并行机 302、304 的此串行层次耦合提供以经压缩字将关于过去事件的信息从第一并行机 302 传送到第二并行机 304 的手段。所传送的信息实际上可为已被第一并行机 302 记录的复杂事件(例如,所关注序列)的概要。

[0039] 图 3 中所展示的并行机 302、304 的两层级层次 300 允许两个独立程序基于同一数

据流而操作。两级层次可类似于建模为不同区的生物大脑中的视觉辨识。在此模型下,所述区实际上为不同型式辨识引擎,每一型式辨识引擎执行类似计算功能(型式匹配)但使用不同程序(签名)。通过将多个并行机 302、304 连接在一起,可获得关于数据流输入的增加的知识。

[0040] 所述层次的第一层级(由第一并行机 302 实施)直接对原始数据流执行处理。也就是说,在输入接口 302A 处接收所述原始数据流且第一并行机 302 的可编程元件可对所述原始数据流做出反应。所述层次的第二层级(由第二并行机 304 实施)处理来自所述第一层级的输出。也就是说,第二并行机 304 在输入接口 304B 处接收来自第一并行机 302 的输出且第二并行机 304 的可编程元件可对第一并行机 302 的输出做出反应。因此,在此实例中,第二并行机 304 不接收所述原始数据流作为输入,而是接收如第一并行机 302 所确定所述原始数据流所匹配的所关注型式的指示。可用辨识来自第一并行机 302 的输出数据流中的型式的 FSM 来编程第二并行机 304。

[0041] 图 4 图解说明两层级层次 400 的另一实例,其中所述层次的一个层级借助多个并行机来实施。此处,层次 400 的第一层级借助三个并行机 402 来实施。将来自三个第一层级并行机 402 中的每一者的输出提供到辨识(例如,识别)来自第一层级并行机 402 的输出中的型式的单个第二层级并行机 404。在其它实例中,可在不同层级处实施不同数目个并行机。每一并行机 402、404 包括数据输入端口 402A、404A、编程接口 402B、404B 及输出端口 402C、404C。

[0042] 图 5 图解说明借助四个并行机 502、504、506 及 508 实施的四层级层次 500,且展示待由每一层级识别的型式的实例。如上文所论述,每一并行机 502、504、506 及 508 包括数据输入端口 502A、504A、506A 及 508A、编程接口 502B、504B、506B 及 508B 以及输出端口 502C、504C、506C 及 508C。四层级层次 500 对应于基于图像中的黑色或白色像素对书面语言的视觉识别。随着层次进展到较高级别,对输入流的积累的知识对应地增长。并行机 502、504、506、508 经级联以实现层次辨识能力。层次 500 的每一连续层级可实施应用于先前层级的经压缩输出的新规则(型式签名)。以此方式,可基于对基本的基元信息的初始检测而识别高度详细的对象。

[0043] 举例来说,到层级 1(第一并行机 502)的原始数据输入流可包含视觉图像的像素信息(例如,不管给定位为黑色/白色、接通/关断)。第一并行机 502 可经编程以识别像素信息所形成的基元型式。举例来说,第一并行机 502 可经配置以识别邻近像素何时形成垂直线、水平线、弧等。此等型式(例如,垂直线、水平线、弧等)中的每一者可由来自第一并行机 502 的相应输出位(或信号)指示。举例来说,当第一并行机 502 识别出至少 3 个位的垂直线时,可在输出字的第一位上将高信号(例如,逻辑‘1’)输出到第二并行机 504。当第一并行机 502 识别出至少 3 个位的水平线时,可在输出字的第二位上将高信号输出到第二并行机 504。

[0044] 第二层级(第二并行机 504)可经编程以识别来自第一并行机 502 的输出信号中的型式。举例来说,第二并行机 504 可经编程以识别第一并行机 502 所识别的基元型式(线、弧等)的组合所形成的型式。第二并行机 504 可经编程以识别水平线与垂直线何时交叉从而形成字母“t”。如上文所提及,第二并行机 504 所实施的 FSM 中的节点对来自第一并行机 502 的输出做出反应。因此通过识别来自第一并行机 502 的输出位中的型式来识别基

元型式的组合。

[0045] 接着将来自第二并行机 504 的输出输入到第三层级（第三并行机 506）中，所述第三层级可从第二并行机 504 所识别的字母的组合识别字。接着第四层级（第四并行机 508）可识别第三并行机 506 所识别的字所形成的短语。因此，较高级可经编程以识别较低层级输出中的型式。另外，较低层级可经编程以识别构成在较高级中所识别的型式的分量。

[0046] 来自像素信息的字母、字及短语的视觉识别用作实例；然而，本文所述的层次方法及设备可应用于其它数据及用于其它用途。举例来说，可对对应于声音的数据使用层次分析以从第一层级处的音素的组合识别音节且从第二层级处的音节的组合识别字。在其它实例中，层次分析可应用于以层次方式在自身上构建的机器数据（例如，原始 0 及 1）。

[0047] 虽然图 5 图解说明层之间的特定及个别连接，但应理解，可实施一层次，在所述层次中将来自一个层级的输出前馈或反馈到所述层次的其它层级。举例来说，来自第二并行机 504 的输出可发送到第四并行机 508，而来自第四并行机 508 的输出可反馈到第三并行机 506。一般来说，层次可经实施以使得来自并行机的检测状态信息馈送到其它并行机中的一者或一者以上或其全部。

[0048] 在一些实例中，在层次结构中使用反馈来更新一个或一个以上层级所使用的程序。举例来说，来自第一层级的输出可提供到第二层级以重新编程所述第二层级。此可用以基于在第一层级中所识别（或未识别）的型式而更新第二层级所应用的规则。在一实例中，第一层级为层次中高于第二层级的层级。举例来说，较低层级可经重新编程以基于较高级所识别的型式而寻找非由程序最初指定的额外型式。在另一实例中，可通知较低层级：较低层级所识别的特定型式为显著的，因为所述特定型式与其它型式组合以形成显著事件。在又一实例中，可通知较低层级：所识别的特定型式不具有特定值且如此较低层级可停止识别所述型式。在一实例中，可随时间执行重新编程，以使得在一时间周期内通过小的调整来递增地修改给定层级的程序。

[0049] 图 6 图解说明四层级层次 600 的实例，其使用反馈来重新编程所述层次的若干部分。借助四个并行机 602、604、606、608 来实施四层级层次 600，每一并行机具有数据输入端口 602A、604A、606A、608A、编程接口 602B、604B、606B、608B 及输出端口 602C、604C、606C、608C。第一并行机 602 实施层次 600 的第一层级且将输出提供到实施层次 600 的第二层级的第二并行机 604。第三并行机 606 及第四并行机 608 同样实施层次 600 的第三层级及第四层级。在一实例中，基于层次 600 对第一并行机 602 所接收的输入数据的分析而将来自第四并行机 608 的输出作为层次 600 的输出发送到外部装置。因此，来自第四并行机 608 的输出对应于层次 600 的集合输出。在其它实例中，来自其它并行机 608 的输出可对应于层次 600 的集合输出。

[0050] 将来自第二并行机 604、第三并行机 606 及第四并行机 608 的输出各自反馈到下面层级处的并行机 602、604、606 的编程接口 602B、604B、606B。举例来说，将来自第四并行机 608 的输出反馈到第三并行机 606 的编程接口 606B 中。因此可基于来自第四并行机 608 的输出而重新编程第三并行机 606。因此，第三并行机 608 可在运行时间期间修改其程序。可分别基于来自第二并行机 604 及第三并行机 606 的输出而在运行时间期间类似地重新编程第一并行机 602 及第二并行机 604。

[0051] 在实例中，来自并行机 604、606、608 的反馈经分析及编译以形成用于重新编程并

行机 602、604、606 的程序（例如，图像）。举例来说，来自并行机 408 的输出在被发送到编程接口 606B 之前由处理装置 614 分析并编译。处理装置 614 可基于来自并行机 608 的输出而产生用于并行机 606 的经更新程序。处理装置 614 可分析所述输出并编译用于第三并行机 606 的经更新程序。接着可通过编程接口 606B 将所述经更新程序加载到第三并行机 606 上以重新编程第三并行机 606。在一实例中，所述经更新程序可仅含有从当前程序的部分改变。因此，在一实例中，经更新程序仅替换并行机 602、604、606、608 上的当前程序的一部分。在另一实例中，经更新程序替换当前程序的全部或一大部分。同样，处理装置 610、612 可基于来自第二并行机 604 及第三并行机 606 的输出而以类似方式分析反馈并编译经更新程序。可借助一个或一个以上额外并行机来实施或可借助不同类型的机器（例如，具有冯·诺伊曼架构的计算机）来实施处理装置 610、612、614。

[0052] 在一些实例中，处理装置 610、612、614 在编译新的程序之前分析来自较高层级的输出。在一实例中，处理装置 610、612、614 分析所述输出以确定如何更新较低层级程序且接着基于所述分析而编译新的（例如，经更新、原始）较低层级程序。虽然在层次 600 中给定并行机处的反馈是从直接在所述给定并行机上面的层级接收的，但反馈可从任一并行机到较高、较低或同一层级处的另一并行机。举例来说，反馈可在并行机的编程输入处从所述同一并行机的输出或从相同、较高或较低层级处的另一并行机的输出接收。另外，并行机可从多个不同并行机接收反馈。基于反馈对并行机的重新编程可与对输入数据中的型式识别在时间上断开（例如，并不随着原始数据的处理实时地）。

[0053] 沿层次向下往回发送信息以影响较低层级的重新编程的目的是可使得较低层级在辨别所关注型式时可变得更高效。沿层次向下发送信息的另一目的是在较低层级中实现较高水平的敏锐度。在一些实例中，认识到将信息传送到层次的较高层级花费时间，因此在可能时避免将信息发送到较高层级的过程。在一些实例中，较高层级可基本上用以解析对于系统来说为新的型式的识别。此可类似于在生物大脑的大脑新皮质中发生的所使用过程。在一实例中，如果可在较低层级处完全解析一型式，那么应这样做。反馈机制为将“学习”传送到层次的较低层级的一种方法。沿层次向下回推信息的此过程将帮助保留层次的上部层级以用于处理新的或不熟悉的型式。此外，可通过减小穿过层次的各个层级的数据传送量来加速整个辨识过程。

[0054] 反馈可使层次的较低层级对输入处的数据流更加敏锐地敏感。此“向下推”信息的结果是可在层次的较低层级处做出决策且所述决策可如此快地完成。因此，在一实例中，来自较低层级并行机（例如，第一并行机 602）的输出可对应于从层次 600 到另一装置的集合输出连同来自第四并行机 608 的输出。外部装置可（例如）监视来自这些并行机 602、608 中的每一者的输出以确定层次 600 何时已识别出型式。

[0055] 在一实例中，反馈信息可包括对应于所分析的数据流的识别信息。举例来说，所述识别信息可包括数据的识别特性、数据的格式、数据的协议及 / 或任一其它类型的识别信息。所述识别信息可由（例如）处理装置 610 收集、分析且用来修改（例如，调适）用于输入数据的分析方法。接着可用经调适的分析方法来编程并行机 100。识别信息可包括（例如）输入数据的语言。并行机 100 可首先经编程以确定输入数据的语言且一旦已识别对应于输入的语言便可在运行时间期间对所述并行机进行调适（例如，重新编程）。用于并行机 100 的经调适分析方法可更特定地对应于用于所识别语言的分析方法。最后，并行机 100 可

使用经调适分析方法来分析未来的输入数据。反馈过程可反复以使得可在输入数据中找到额外识别信息以允许进一步调适分析方法。

[0056] 用于加载到并行机 100 上的程序（例如，图像）可由下文关于图 12 所论述的编译器产生。一般来说，编译可为计算密集的过程，且可能在第一次编译型式签名的大数据库时最明显。在运行时间操作中，较高层级的并行机 100 可正在以用于较低层级并行机的增量程序更新的形式将反馈提供到较低层级。因此，到较低层级并行机的反馈信息可为对原始程序的编译起来较不计算密集的小得多的增量更新。

[0057] 图 7 图解说明借助四个并行机 702、704、706、708 实施的四层级层次 700 的另一实例。四个并行机 702、704、706、708 各自具有数据输入端口 702A、704A、706A、708A、编程接口 702B、704B、706B、708B 及输出端口 702C、704C、706C、708C。另外，在一些实例中，四层级层次 700 可包括处理装置 710、712、714 以编译用于并行机 702、704 及 706 的程序。在四层级层次 700 中，第二并行机 704、第三并行机 706 及第四层级并行机 708 从较低层级并行机 702、704、706 的输出接收输入数据且从原始数据流接收输入数据。因此，层级 2、3 及 4 可来自较低层级与原始数据的型式的组合识别型式。

[0058] 如从图 6 及 7 可见，并行机 100 可以几乎任一方式级联，其中可将到层次的原始数据输入以及来自并行机 100 的输出发送到任一其它并行机 100，包括其自身。此外，可将来自给定并行机 100 的输出作为到数据输入端口 110 中的输入数据及 / 或作为用于更新并行机 100 的程序的反馈发送到另一并行机 100。

[0059] 由于并行机 100 处理输入数据的一个数据循环（例如，一位、一字）的时间，串行地级联并行机 100 可增加通过所有并行机 100 完全处理输入数据流以产生层次的集合输出的时间。由于层次的较低层级可接收较低（最细微的）层级的输入，因此应预期较低层级比高层级的输出更活跃。也就是说，层次中的每一连续层级可汇编较高级对象。在一实例中，并行机 100 具有限制可多快地将输入数据馈送到并行机 100 的最大输入速率。可将此输入速率视为单个数据循环。在每一连续数据循环上，并行机 100 具有激活许多终端节点的可能性。此可致使并行机 100（尤其在层次的较低层级处）产生显著量的输出数据。举例来说，如果将输入作为字节流提供到最低层级并行机 100，那么在任一给定数据循环上，并行机 100 可产生多个字节的结果信息。如果一个字节的信息可产生多个字节的信息，那么并行机 100 的整个层次应同步以使得信息沿层次向上传递。在一些实例中，反馈无需同步。然而，在较低层级处更快地接收反馈，所述较低层级便可更快地调适且分析更高效。

[0060] 作为一实例，层次（借助单个并行机 100 实施）的每一层级的最大大小输出可等于 1024 个字节且层次的深度可等于 4 个层级。用于并行机 100 的输入数据流数据速率可等于 128MB/秒。在这些条件下，可在 7.63 微秒中横越层次的每一层级。在四层级层次的情况下，并行机 100 的整个堆叠的总稳定时间将为 7.63 微秒的 4 倍或 30.5 微秒。在 30.5 微秒的稳定时间的情况下，暗示输入数据频率应限于 32KB/s。

[0061] 值得注意地，此高度取决于并行机 100 的配置。并行机 100 可为可配置的以对输入数据速率对状态机大小进行折衷。另外，如果对产生加载到并行机 100 上的个别图像的编译器做出对应修改，那么可调整到所述并行机的输入字大小。

[0062] 在一实例中，可在具有冯·诺伊曼架构的机器上以软件来实施上文所述的层次结构。因此，软件指令可致使处理器对原始数据实施第一层级分析 FSM。来自第一层级 FSM 的

输出接着可由所述处理器使用第二层级分析来处理,等等。此外,上文所论述的反馈循环可经实施以使得基于(例如)第二层级分析的输出而修改第一层级分析。

[0063] 图 8 到 11 图解说明在本文中称作“FSM 引擎 800”的并行机的实例。在一实例中,FSM 引擎 800 包含有限状态机的硬件实施方案。因此,FSM 引擎 800 实施对应于 FSM 中的多个状态的多个可选择性耦合的硬件元件(例如,可编程元件)。类似于 FSM 中的状态,硬件元件可分析输入流并基于所述输入流而激活下游硬件元件。

[0064] FSM 引擎 800 包括多个可编程元件,包括通用元件及专用元件。所述通用元件可经编程以实施许多不同功能。这些通用元件包括以层次方式组织成行 806(图 9 及 10 中所展示)及块 802(图 8 及 9 中所展示)的 SME 804、805(图 11 中所展示)。为了在以层次方式组织的 SME 804、805 之间路由信号,使用可编程开关的层次,其包括块间开关 803(图 8 及 9 中所展示)、块内开关 808(图 9 及 10 中所展示)及行内开关 812(图 10 中所展示)。SME 804、805 可对应于 FSM 引擎 800 所实施的 FSM 的状态。可通过使用下文所述的可编程开关将 SME 804、805 耦合在一起。因此,可通过将 SME 804、805 编程为对应于状态的功能且通过选择性地将 SME 804、805 耦合在一起以对应于 FSM 中的状态之间的转变而在 FSM 引擎 800 上实施 FSM。

[0065] 图 8 图解说明实例性 FSM 引擎 800 的总体视图。FSM 引擎 800 包括可选择性地与可编程块间开关 803 耦合在一起的多个块 802。另外,块 802 可选择性地耦合到输入块 809(例如,数据输入端口)以用于接收信号(例如,数据)并将数据提供到块 802。块 802 还可选择性地耦合到输出块 813(例如,输出端口)以用于将信号从块 802 提供到外部装置(例如,另一 FSM 引擎 800)。FSM 引擎 800 还可包括编程接口 811 以将程序(例如,图像)加载到 FSM 引擎 800 上。所述图像可编程(例如,设定)SME 804、805 的状态。也就是说,所述图像可将 SME 804、805 配置为以某一方式对输入块 809 处的给定输入做出反应。举例来说,SME 804 可经设定以当在输入块 809 处接收到字符‘a’时输出高信号。

[0066] 在一实例中,可将输入块 809、输出块 813 及/或编程接口 811 实施为寄存器以使得向所述寄存器的写入将数据提供到相应元件或从所述相应元件提供数据。因此,可将来自存储于对应于编程接口 811 的寄存器中的图像的位加载于 SME 804、805 上。虽然图 8 图解说明块 802、输入块 809、输出块 813 与块间开关 803 之间的某一数目个导体(例如,导线、迹线),但应理解在其它实例中可使用更少或更多导体。

[0067] 图 9 图解说明块 802 的实例。块 802 可包括可选择性地与可编程块内开关 808 耦合在一起的多个行 806。另外,行 806 可借助块间开关 803 选择性地耦合到另一块 802 内的另一行 806。在一实例中,包括缓冲器 801 以控制去往/来自块间开关 803 的信号的计时。行 806 包括组织成元件对(本文中称作两个一群组(GOT)810)的多个 SME 804、805。在一实例中,块 802 包含十六(16)个行 806。

[0068] 图 10 图解说明行 806 的实例。GOT 810 可通过可编程行内开关 812 选择性地耦合到行 806 内的其它 GOT 810 及任何其它元件 824。GOT 810 还可借助块内开关 808 耦合到其它行 806 中的其它 GOT 810,或借助块间开关 803 耦合到其它块 802 中的其它 GOT 810。在一实例中,GOT 810 具有第一输入 814 及第二输入 816 以及输出 818。第一输入 814 耦合到 GOT 810 的第一 SME 804 且第二输入 816 耦合到 GOT 810 的第二 SME 804。

[0069] 在一实例中,行 806 包括第一多个行互连导体 820 及第二多个行互连导体 822。在

一实例中, GOT 810 的输入 814、816 可耦合到一个或一个以上行互连导体 820、822, 且输出 818 可耦合到一个行互连导体 820、822。在一实例中, 第一多个行互连导体 820 可耦合到行 806 内的每一 GOT 810 的每一 SME 804。第二多个行互连导体 822 可耦合到行 806 内的每一 GOT 810 的一个 SME 804, 但不可耦合到 GOT 810 的另一 SME 804。在一实例中, 第二多个行互连导体 822 的第一半可耦合到行 806 内的 SME 804 的第一半 (来自每一 GOT 810 的一个 SME 804) 且第二多个行互连导体 822 的第二半可耦合到行 806 内的 SME 804 的第二半 (来自每一 GOT 810 的另一 SME 804)。第二多个行互连导体 822 与 SME 804、805 之间的有限连接性在本文中称作“奇偶”。在一实例中, 行 806 还可包括专用元件 824, 例如计数器、可编程布尔逻辑元件、现场可编程门阵列 (FPGA)、专用集成电路 (ASIC)、可编程处理器 (例如, 微处理器) 及其它元件。

[0070] 在一实例中, 专用元件 824 包括计数器 (在本文中也称作计数器 824)。在一实例中, 计数器 824 包含 12 位可编程倒数计数器。12 位可编程计数器 824 具有计数输入、复位输入及零计数输出。计数输入在被断言时使计数器 824 的值递减 1。复位输入在被断言时致使计数器 824 从相关联寄存器加载初始值。对于 12 位计数器 824 来说, 可加载多达 12 位的数值作为所述初始值。当计数器 824 的值递减到零 (0) 时, 断言零计数输出。计数器 824 还具有至少两种模式: 脉冲及保持。当将计数器 824 设定为脉冲模式时, 在计数器 824 递减到零时于时钟循环期间断言零计数输出, 且在下一时钟循环处不再断言零计数输出。当将计数器 824 设定为保持模式时, 在计数器 824 递减到零时于时钟循环期间断言零计数输出, 且保持断言所述零计数输出直到计数器 824 由正断言的复位输入复位为止。在一实例中, 专用元件 824 包括布尔逻辑。在一些实例中, 此布尔逻辑可用以从 FSM 引擎 800 中的终端状态 SME 提取信息。所提取的信息可用以将状态信息传送到其它 FSM 引擎 800 及 / 或传送到以重新编程 FSM 引擎 800 或重新编程另一 FSM 引擎 800 的编程信息。

[0071] 图 11 图解说明 GOT 810 的实例。GOT 810 包括具有输入 814、816 且使其输出 826、828 耦合到“或”门 830 及 3 对 1 多路复用器 842 的第一 SME 804 及第二 SME 805。3 对 1 多路复用器 842 可经设定以将 GOT 810 的输出 818 耦合到第一 SME 804、第二 SME 805 或“或”门 830。“或”门 830 可用以将输出 826、828 两者耦合在一起以形成 GOT 810 的共用输出 818。在一实例中, 如上文所论述, 第一 SME 804 及第二 SME 805 展现出奇偶, 其中第一 SME 804 的输入 814 可耦合到行互连导体 822 中的一些行互连导体且第二 SME 805 的输入 816 可耦合到其它行互连导体 822。在一实例中, 可通过设定开关 840 中的任一者或两者而使 GOT 810 内的两个 SME 804、805 级联及 / 或循环回到其自身。可通过将 SME 804、805 的输出 826、828 耦合到其它 SME 804、805 的输入 814、816 而使 SME 804、805 级联。可通过将输出 826、828 耦合到其自己的输入 814、816 而使 SME 804、805 循环回到其自身。因此, 第一 SME 804 的输出 826 可不耦合到第一 SME 804 的输入 814 及第二 SME 805 的输入 816 中的任一者、耦合到其中的一者或耦合到其中的两者。

[0072] 在一实例中, 状态机元件 804、805 包含并行耦合到检测线 834 的多个存储器单元 832, 例如通常用于动态随机存取存储器 (DRAM) 中的那些存储器单元。一个此种存储器单元 832 包含可设定为一数据状态 (例如对应于高值或低值 (例如, 1 或 0) 的数据状态) 的存储器单元。存储器单元 832 的输出耦合到检测线 834 且到存储器单元 832 的输入基于数据流线 836 上的数据而接收信号。在一实例中, 数据流线 836 上的输入经解码以选择存储

器单元 832 中的一者。选定存储器单元 832 将其所存储的数据状态作为输出提供到检测线 834 上。举例来说,可将在数据输入端口 809 处所接收的数据提供到解码器(未展示)且所述解码器可选择数据流线 836 中的一者。在一实例中,所述解码器可将 ASCII 字符转换成 256 个位中的 1。

[0073] 因此,当存储器单元 832 设定为高值且数据流线 836 上的数据对应于存储器单元 832 时,存储器单元 832 将高信号输出到检测线 834。当数据流线 836 上的数据对应于存储器单元 832 且存储器单元 832 设定为低值时,存储器单元 832 将低信号输出到检测线 834。检测线 834 上来自存储器单元 832 的输出由检测电路 838 感测。在一实例中,输入线 814、816 上的信号将相应检测电路 838 设定为活动或不活动状态。当设定为不活动状态时,不管相应检测线 834 上的信号如何,检测电路 838 均在相应输出 826、828 上输出低信号。当设定为活动状态时,检测电路 838 在从相应 SME 804、805 的存储器单元 834 中的一者检测到高信号时在相应输出线 826、828 上输出高信号。当处于活动状态中时,检测电路 838 在来自相应 SME 804、805 的所有存储器单元 834 的信号为低时在相应输出线 826、828 上输出低信号。

[0074] 在一实例中,SME 804、805 包括 256 个存储器单元 832 且每一存储器单元 832 耦合到不同数据流线 836。因此,SME 804、805 可经编程以在数据流线 836 中的选定一者或一者以上在其上具有高信号时输出高信号。举例来说,SME 804 可将第一存储器单元 832(例如,位 0)设定为高且将所有其它存储器单元 832(例如,位 1 到 255)设定为低。当相应检测电路 838 处于活动状态中时,SME 804 在对应于位 0 的数据流线 836 在其上具有高信号时在输出 826 上输出高信号。在其它实例中,SME 804 可经设定以通过将适当存储器单元 832 设定为高值而在多个数据流线 836 中的一者在其上具有高信号时输出高信号。

[0075] 在一实例中,可通过从相关联寄存器读取位而将存储器单元 832 设定为高值或低值。因此,可通过将编译器所创建的图像存储到寄存器中并将所述寄存器中的位加载到相关联存储器单元 832 中来编程 SME 804。在一实例中,所述编译器所创建的图像包括高与低(例如,1 与 0)位的二进制图像。所述图像可编程 FSM 引擎 800 以通过级联 SME 804、805 而作为 FSM 操作。举例来说,可通过将检测电路 838 设定为活动状态而将第一 SME804 设定为活动状态。第一 SME 804 可经设定以在对应于位 0 的数据流线 836 在其上具有高信号时输出高信号。第二 SME 805 最初可设定为不活动状态,但可经设定以在活动时在对应于位 1 的数据流线 836 在其上具有高信号时输出高信号。可通过设定第一 SME804 的输出 826 以耦合到第二 SME 805 的输入 816 来级联第一 SME 804 与第二 SME 805。因此,当在对应于位 0 的数据流线 836 上感测到高信号时,第一 SME 804 在输出 826 上输出高信号且将第二 SME 805 的检测电路 838 设定为活动状态。当在对应于位 1 的数据流线 836 上感测到高信号时,第二 SME 805 在输出 828 上输出高信号以激活另一 SME805 或供从 FSM 引擎 800 输出。

[0076] 图 10 图解说明用于编译器将源代码转换成经配置以编程并行机的图像的方法 1000 的实例。方法 1000 包括将源代码剖析成语法树(框 1002),将所述语法树转换成自动机(框 1004),优化所述自动机(框 1006),将所述自动机转换成网表(框 1008),将所述网表置于硬件上(框 1010),路由所述网表(框 1012)及公布所得图像(框 1014)。

[0077] 在一实例中,编译器包括允许软件开发者创建用于在 FSM 引擎 800 上实施 FSM 的图像的应用程序编程接口(API)。编译器提供用以将源代码中的输入正则表达式集转换成

经配置以编程 FSM 引擎 800 的图像的方法。可通过用于具有冯·诺伊曼架构的计算机的指令来实施所述编译器。这些指令可致使计算机上的处理器实施编译器的功能。举例来说,所述指令在由处理器执行时可致使处理器对所述处理器可存取的源代码执行框 1002、1004、1006、1008、1010、1012 及 1014 中所述的动作。图 13 中展示具有冯·诺伊曼架构的实例性计算机且下文对其进行描述。

[0078] 在一实例中,源代码描述用于识别符号群组内的符号型式的搜索串。为了描述搜索串,源代码可包括多个正则表达式(regex)。正则表达式可为用于描述符号搜索型式的串。正则表达式广泛地用于各种计算机领域中,例如程序设计语言、文本编辑器、网络安全及其它领域。在一实例中,编译器所支持的正则表达式包括用于搜索未结构化数据的搜索准则。未结构化数据可包括自由形式的数据且不具有应用于所述数据内的字的索引。字可包括所述数据内的可打印及不可打印的字节的任一组合。在一实例中,编译器可支持多种不同源代码语言以用于实施包括 Perl(例如,Perl 兼容正则表达式(PCRE))、PHP、Java 及 .NET 语言的正则表达式。

[0079] 在框 1002 处,编译器可剖析源代码以形成关系连接的运算符的布置,其中不同类型的运算符对应于源代码所实施的不同函数(例如,源代码中的正则表达式所实施的不同函数)。剖析源代码可创建所述源代码的类属表示。在一实例中,所述类属表示包含源代码中的正则表达式的经编码表示,其呈称作语法树的树形图的形式。本文所述的实例涉及作为语法树(还称作“抽象语法树”)的布置,然而在其它实例中可使用具体语法树或其它布置。

[0080] 如上文所提及,由于编译器可支持源代码的多种语言,因此不管语言如何剖析均将源代码转换成非语言特定表示(例如,语法树)。因此,不管源代码的语言如何,由编译器进行的进一步处理(框 1004、1006、1008、1010)均可从共用输入结构起作用。

[0081] 如上所述,语法树包括关系连接的多个运算符。语法树可包括多种不同类型的运算符。也就是说,不同运算符可对应于源代码中的正则表达式所实施的不同函数。

[0082] 在框 1004 处,将语法树转换成自动机。自动机包含 FSM 的软件模型且可因此分类为确定性或非确定性。确定性自动机在给定时间具有单个执行路径,而非确定性自动机具有多个同时执行路径。所述自动机包含多个状态。为了将语法树转换成自动机,将语法树中的运算符及运算符之间的关系转换成状态,其中所述状态之间具有转变。在一实例中,可部分地基于 FSM 引擎 800 的硬件而转换所述自动机。

[0083] 在一实例中,用于自动机的输入符号包括字母、数字 0 到 9 及其它可打印字符的符号。在一实例中,输入符号由字节值 0 到 255(包括 0 及 255)表示。在一实例中,自动机可表示为有向图,其中所述图的节点对应于状态集。在一实例中,输入符号 α (即, $\delta(p, \alpha)$) 上从状态 p 到状态 q 的转变由从节点 p 到节点 q 的有向连接展示。在一实例中,自动机的反转产生新的自动机,其中某一符号 α 上的每一转变 $p \rightarrow q$ 在同一符号上反转 $q \rightarrow p$ 。在反转中,开始状态变为最终状态且最终状态变为开始状态。在一实例中,自动机所接受(例如,匹配)的语言是当依序输入到所述自动机中时将到达最终状态的所有可能字符串的集。所述自动机所接受的语言中的每一串追踪从开始状态到一个或一个以上最终状态的路径。

[0084] 在框 1006 处,在构造自动机之后,优化所述自动机以除其它之外还减小其复杂度及大小。可通过组合冗余状态来优化所述自动机。

[0085] 在框 1008 处,将经优化的自动机转换成网表。将所述自动机转换成网表将所述自动机的每一状态映射到 FSM 引擎 800 上的硬件元件(例如,SME 804、805、其它元件 824)并确定所述硬件元件之间的连接。

[0086] 在框 1010 处,放置所述网表以选择目标装置的对应于所述网表的每一节点的特定硬件元件(例如,SME 804、805、专用元件 824)。在一实例中,放置基于 FSM 引擎 800 的一般输入及输出约束而选择每一特定硬件元件。

[0087] 在框 1012 处,对所放置的网表进行路由以确定用于可编程开关(例如,块间开关 803、块内开关 808 及行内开关 812)的设定,以便将选定硬件元件耦合在一起以实现网表所描述的连接。在一实例中,通过确定 FSM 引擎 800 的将用以连接选定硬件元件及用于可编程开关的设定的特定导体来确定用于可编程开关的设定。相比于框 1010 处的放置,路由可能考虑硬件元件之间的连接的更特定限制。因此,假定有对 FSM 引擎 800 上的导体的实际限制,路由可调整通过全域放置所确定的所述硬件元件中的一些硬件元件的位置以便进行适当连接。

[0088] 一旦网表经放置及路由,便可将所述经放置及路由的网表转换成用于编程 FSM 引擎 800 的多个位。所述多个位在本文中称作图像。

[0089] 在框 1014 处,编译器公布图像。所述图像包含用于编程 FSM 引擎 800 的特定硬件元件及/或可编程开关的多个位。在其中所述图像包含多个位(例如,0 及 1)的实施例中,所述图像可称作二进制图像。可将所述位加载到 FSM 引擎 800 上以编程 SME 804、805、专用元件 824 及可编程开关的状态,以使得经编程 FSM 引擎 800 实施具有源代码所描述的功能性的 FSM。放置(框 1010)及路由(框 1012)可将 FSM 引擎 800 中的特定位置处的特定硬件元件映射到自动机中的特定状态。因此,所述图像中的位可编程特定硬件元件及/或可编程开关以实施所要功能。在一实例中,可通过将机器代码保存到计算机可读媒体来公布所述图像。在另一实例中,可通过将所述图像显示于显示装置上来公布所述图像。在又一实例中,可通过将图像发送到另一装置(例如用于将图像加载到 FSM 引擎 800 上的编程装置)来公布所述图像。在又一实例中,可通过将图像加载到并行机(例如,FSM 引擎 800)上来公布所述图像。

[0090] 在一实例中,可通过将来自图像的位值直接加载到 SME 804、805 及其它硬件元件 824 或通过将所述图像加载到一个或一个以上寄存器中且接着将来自所述寄存器的位值写入到 SME 804、805 及其它硬件元件 824 而将所述图像加载到 FSM 引擎 800 上。在一实例中,可编程开关(例如,块间开关 803、块内开关 808 及行内开关 812)的状态。在一实例中,FSM 引擎 800 的硬件元件(例如,SME 804、805、其它元件 824、可编程开关 803、808、812)经存储器映射以使得编程装置及/或计算机可通过将所述图像写入到一个或一个以上存储器地址而将所述图像加载到 FSM 引擎 800 上。

[0091] 本文所述的方法实例可为至少部分地机器或计算机实施的。一些实例可包括用指令编码的计算机可读媒体或机器可读媒体,所述指令可操作以配置电子装置以执行以上实例中所述的方法。这些方法的实施方案可包括代码,例如微码、汇编语言代码、高级语言代码等。此代码可包括用于执行各种方法的计算机可读指令。所述代码可形成计算机程序产品的部分。此外,所述代码可在执行期间或在其它时间有形地存储于一个或一个以上易失性或非易失性计算机可读媒体上。这些计算机可读媒体可包括(但不限于)硬盘、可装卸

磁盘、可装卸光盘（例如，压缩光盘及数字视频光盘）、盒式磁带、存储卡或存储棒、随机存取存储器（RAM）、只读存储器（ROM）等。

[0092] 图 13 大体图解说明具有冯·诺伊曼架构的计算机 1500 的实例。在阅读并理解本发明的内容后，所属领域的技术人员将即刻理解可从基于计算机的系统中的计算机可读媒体启动软件程序以执行所述软件程序中所界定的功能的方式。所属领域的技术人员将进一步理解可用来创建经设计以实施及执行本文所揭示的方法的一个或一个以上软件程序的各种程序设计语言。可使用面向对象的语言（例如 Java、C++ 或一种或一种以上其它语言）以面向对象的格式来结构化程序。或者，可使用程序语言（例如汇编、C 等）以面向程序的格式来结构化程序。软件组件可使用所属领域的技术人员众所周知的若干种机制（例如应用程序接口或进程间通信技术，包括远程程序呼叫或其它）中的任一者进行通信。各种实施例的教示不限于任一特定程序设计语言或环境。

[0093] 因此，可实现其它实施例。举例来说，制品（例如计算机、存储器系统、磁盘或光盘、某一其它存储装置或任一类型的电子装置或系统）可包括耦合到其上存储有指令 1524（例如，计算机程序指令）的计算机可读媒体 1522（例如存储器（例如，可装卸存储媒体以及包括电、光学或电磁导体的任一存储器））的一个或一个以上处理器 1502，所述指令在由一个或一个以上处理器 1502 执行时导致执行关于以上方法所述的动作中的任一者。

[0094] 计算机 1500 可采取具有直接及 / 或使用总线 1508 耦合到若干个组件的处理器 1502 的计算机系统的形式。此些组件可包括主存储器 1504、静态或非易失性存储器 1506 及大容量存储装置 1516。耦合到处理器 1502 的其它组件可包括输出装置 1510（例如视频显示器）、输入装置 1512（例如键盘）及光标控制装置 1514（例如鼠标）。用以将处理器 1502 及其它组件耦合到网络 1526 的网络接口装置 1520 还可耦合到总线 1508。可利用若干个众所周知的传送协议（例如，HTTP）中的任一者经由网络接口装置 1520 在网络 1526 上进一步传输或接收指令 1524。耦合到总线 1508 的这些元件中的任一者可取决于待实现的特定实施例而不存在、单独存在或以复数数目存在。

[0095] 在一实例中，处理器 1502、存储器 1504、1506 或存储装置 1516 中的一者或一者以上可各自包括在执行时可致使计算机 1500 执行本文所述的方法中的任何一者或一者以上的指令 1524。在替代实施例中，计算机 1500 作为独立装置操作或可连接（例如，连网）到其它装置。在连网环境中，计算机 1500 可在服务器 - 客户端网络环境中以服务器或客户端装置的资格或在対等（或分布式）网络环境中作为对等装置操作。计算机 1500 可包括个人计算机（PC）、平板 PC、机顶盒（STB）、个人数字助理（PDA）、蜂窝式电话、网络器具、网络路由器、交换机或桥接器或能够执行指定待由所述装置采取的动作的指令集（顺序或以其它方式）的任一装置。此外，尽管仅图解说明单个计算机 1500，但术语“计算机”还应视为包括个别或共同地执行一指令集（或多个指令集）以执行本文所论述的方法中的任何一者或一者以上的任一装置集合。

[0096] 计算机 1500 还可包括输出控制器 1528 以用于使用一个或一个以上通信协议（例如，通用串行总线（USB）、IEEE 1394 等）与外围装置通信。输出控制器 1528 可（例如）将图像提供到以通信方式耦合到计算机 1500 的编程装置 1530。编程装置 1530 可经配置以编程并行机（例如，并行机 100、FSM 引擎 800）。在其它实例中，编程装置 1530 可与计算机 1500 集成在一起并耦合到总线 1508 或可经由网络接口装置 1520 或另一装置与计算机

1500 通信。

[0097] 尽管将计算机可读媒体 1524 展示为单个媒体,但术语“计算机可读媒体”应视为包括存储一个或一个以上指令集 1524 的单个媒体或多个媒体(例如,集中式或分布式数据库,或相关联高速缓冲存储器及服务器,及/或各种存储媒体,例如处理器 1502 寄存器、存储器 1504、1506 及存储装置 1516)。术语“计算机可读媒体”还应视为包括能够存储、编码或载运指令集以供所述计算机执行且致使所述计算机执行本发明的方法中的任何一者或一者以上或能够存储、编码或载运此指令集所利用或与此指令集相关联的数据结构的任一媒体。术语“计算机可读媒体”因此应视为包括(但不限于)有形媒体(例如固态存储器)、光学媒体及磁性媒体。

[0098] 提供发明摘要以符合 37C.F.R. 第 1.72(b) 款,其需要将允许读者探知所述技术揭示内容的本质及主旨的摘要。提交本摘要基于以下理解:其并非将用以限制或解释权利要求书的范围或含义。特此将所附权利要求书并入到详细描述中,其中每一权利要求自身作为单独实施例。

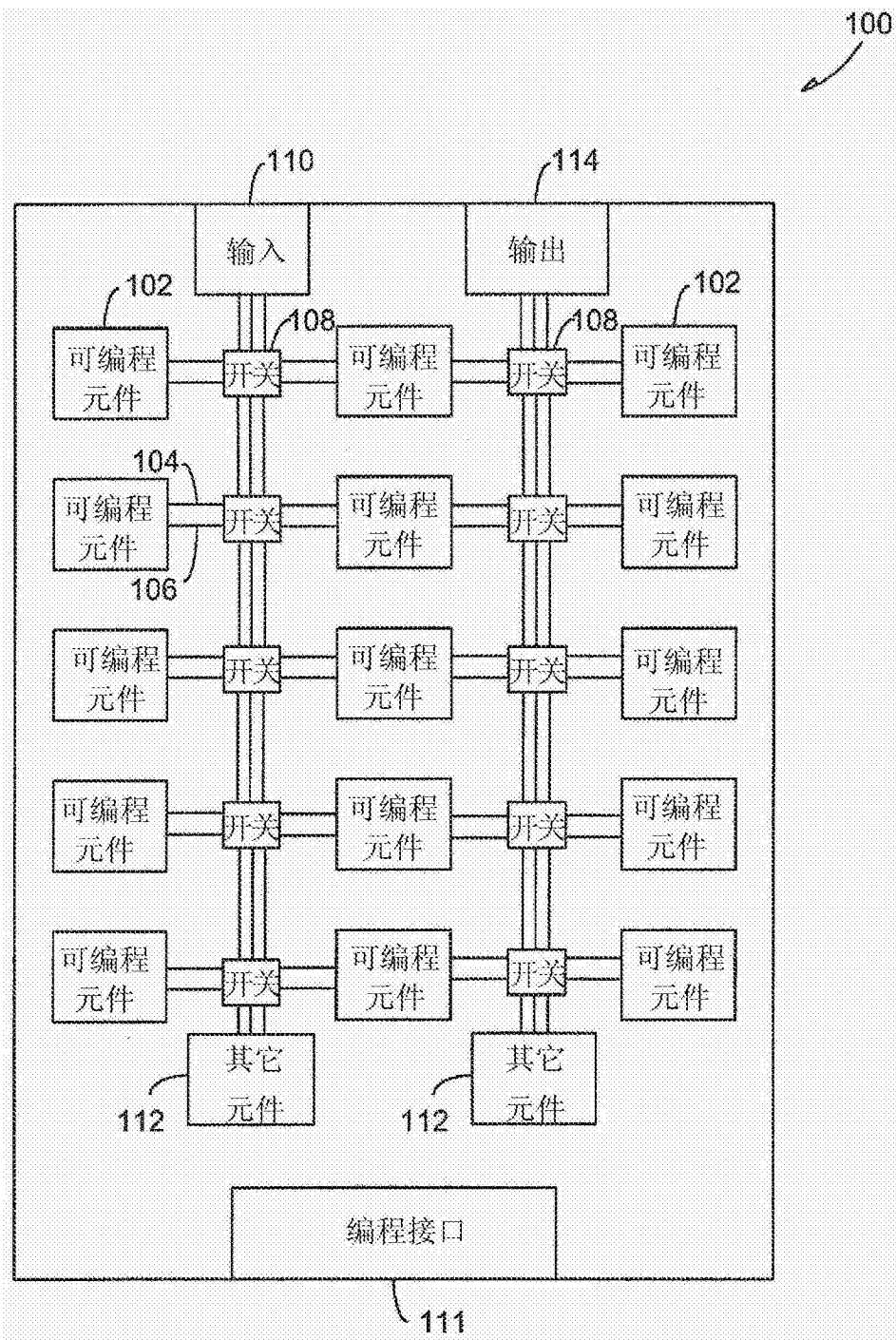


图 1

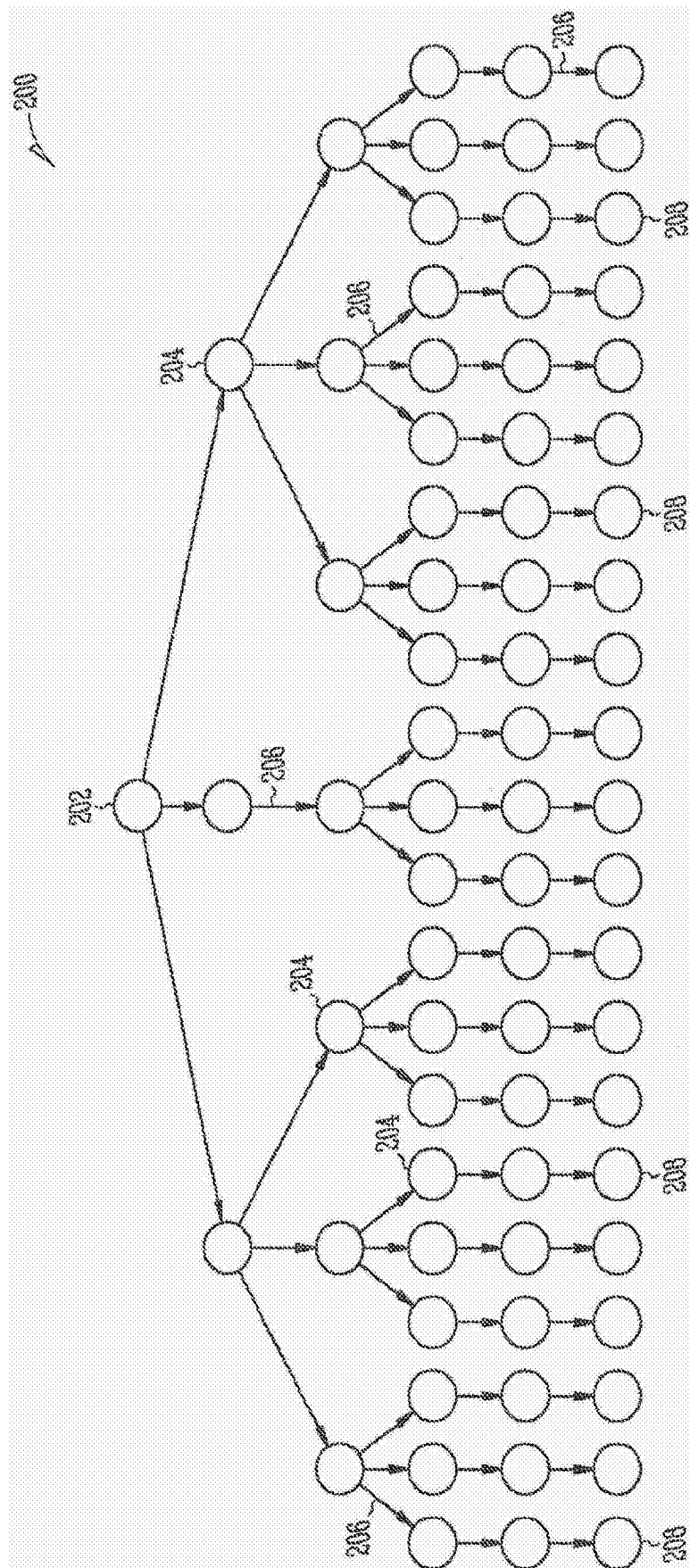


图 2

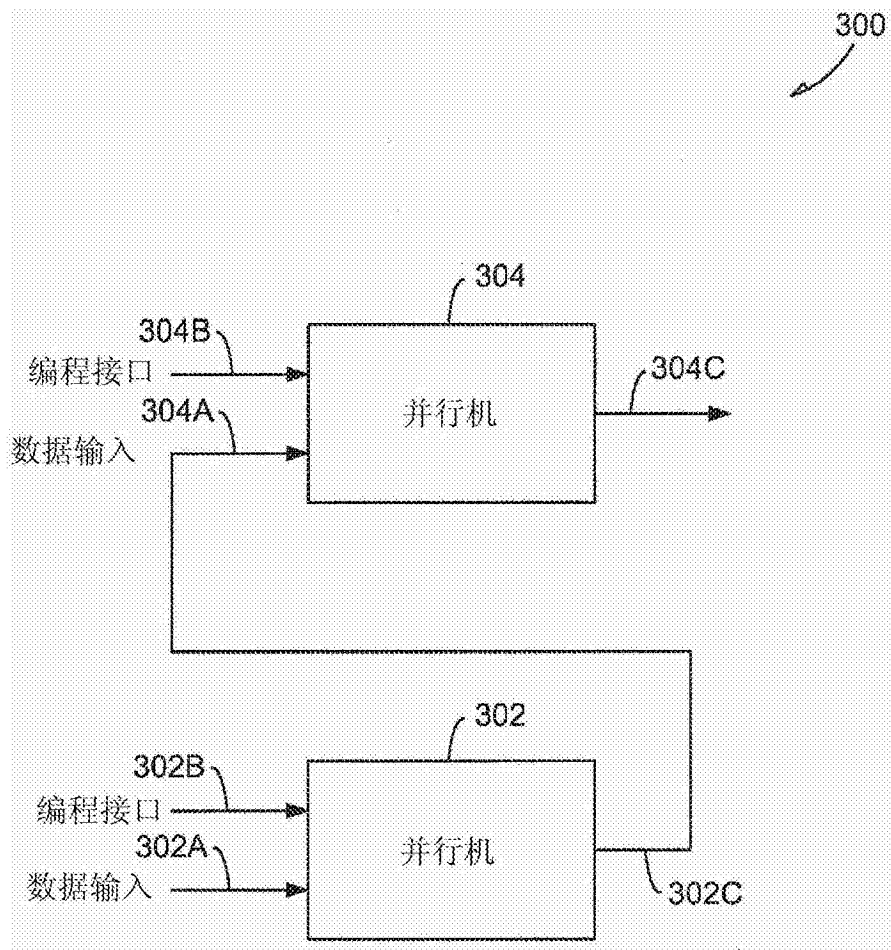


图 3

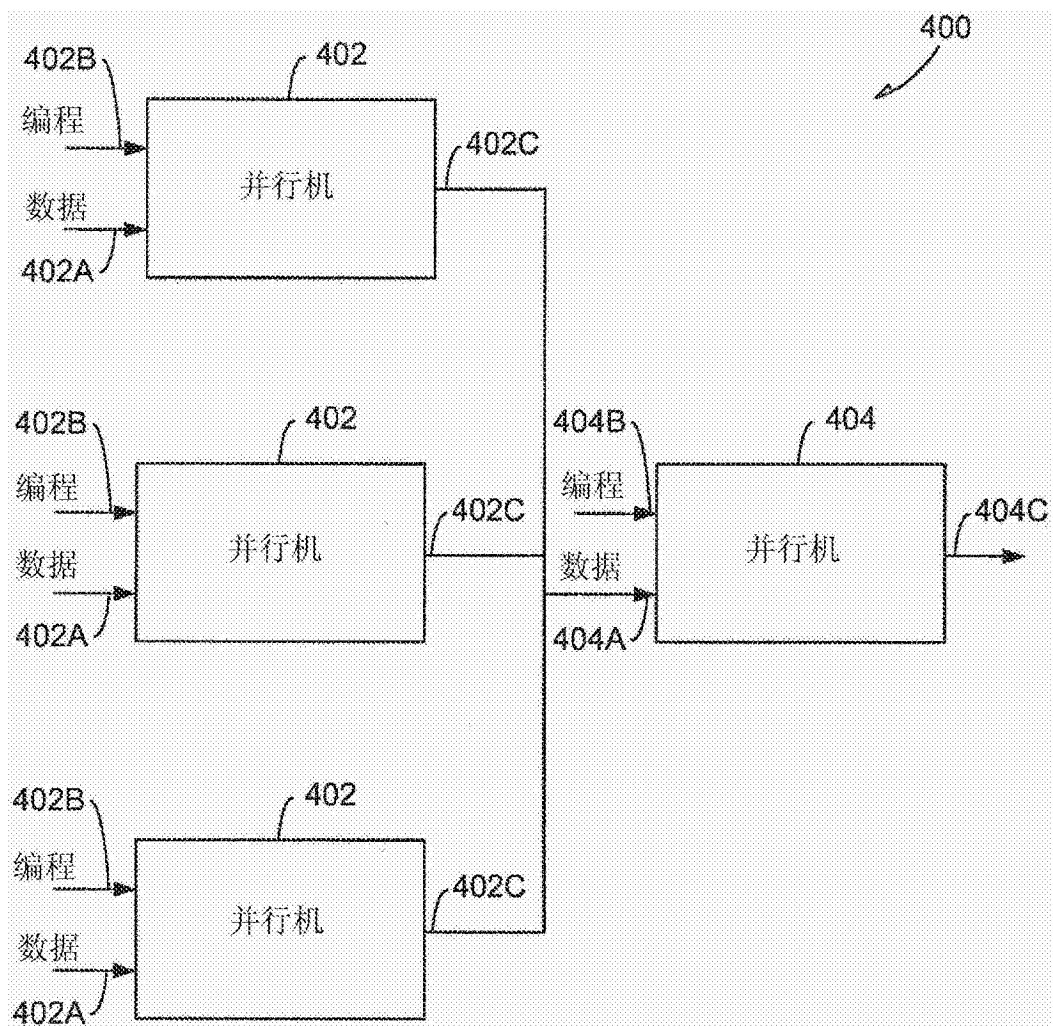


图 4

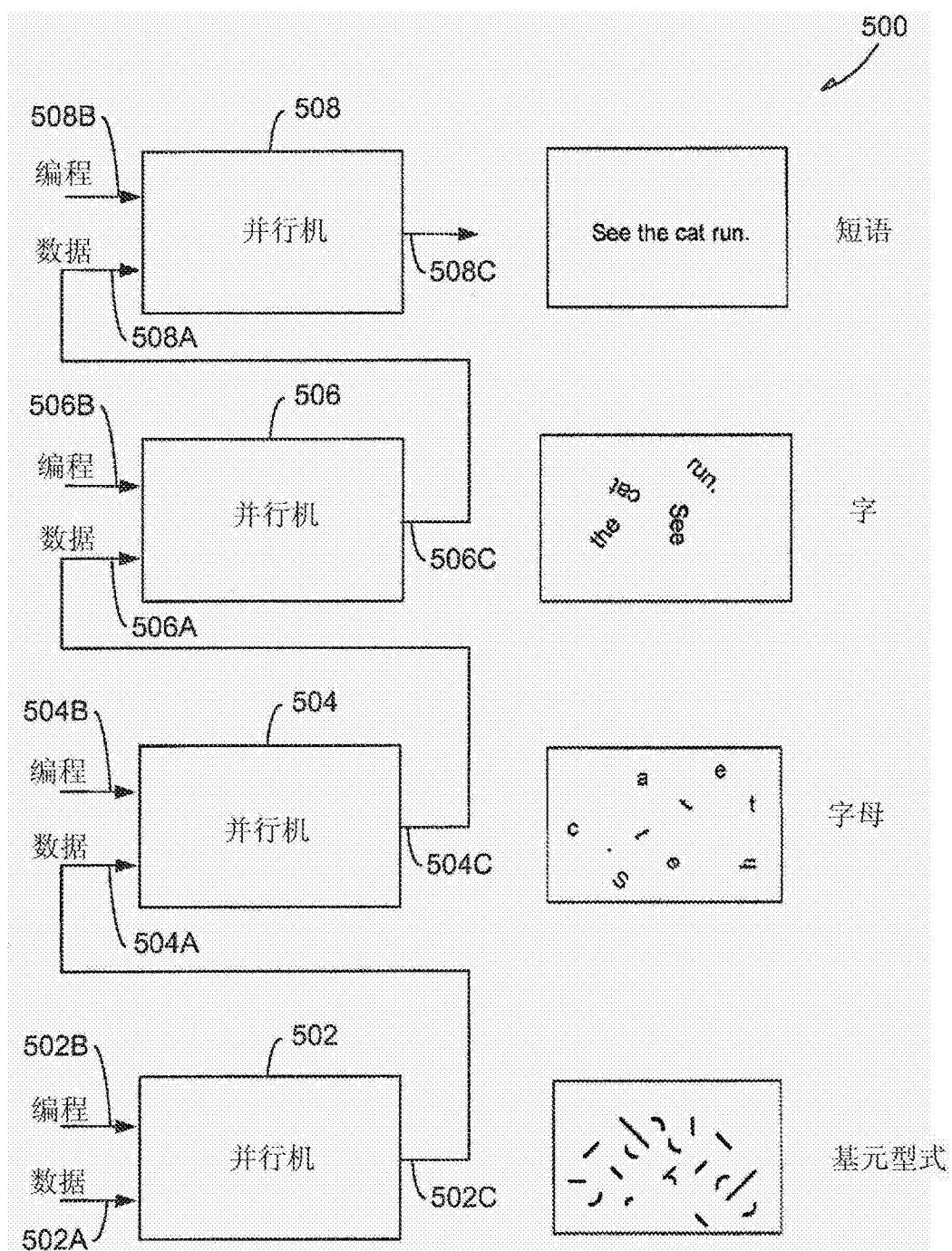


图 5

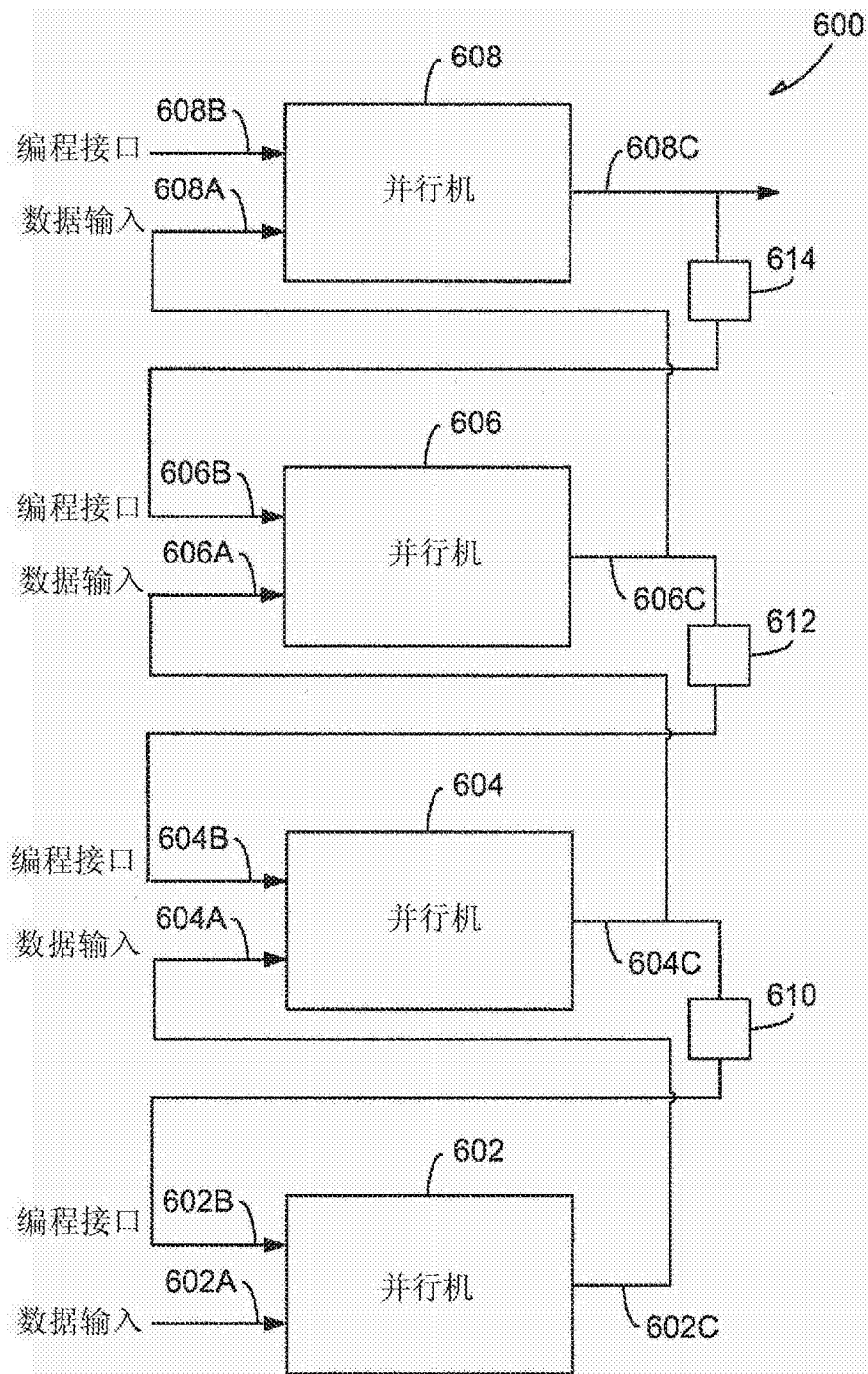


图 6

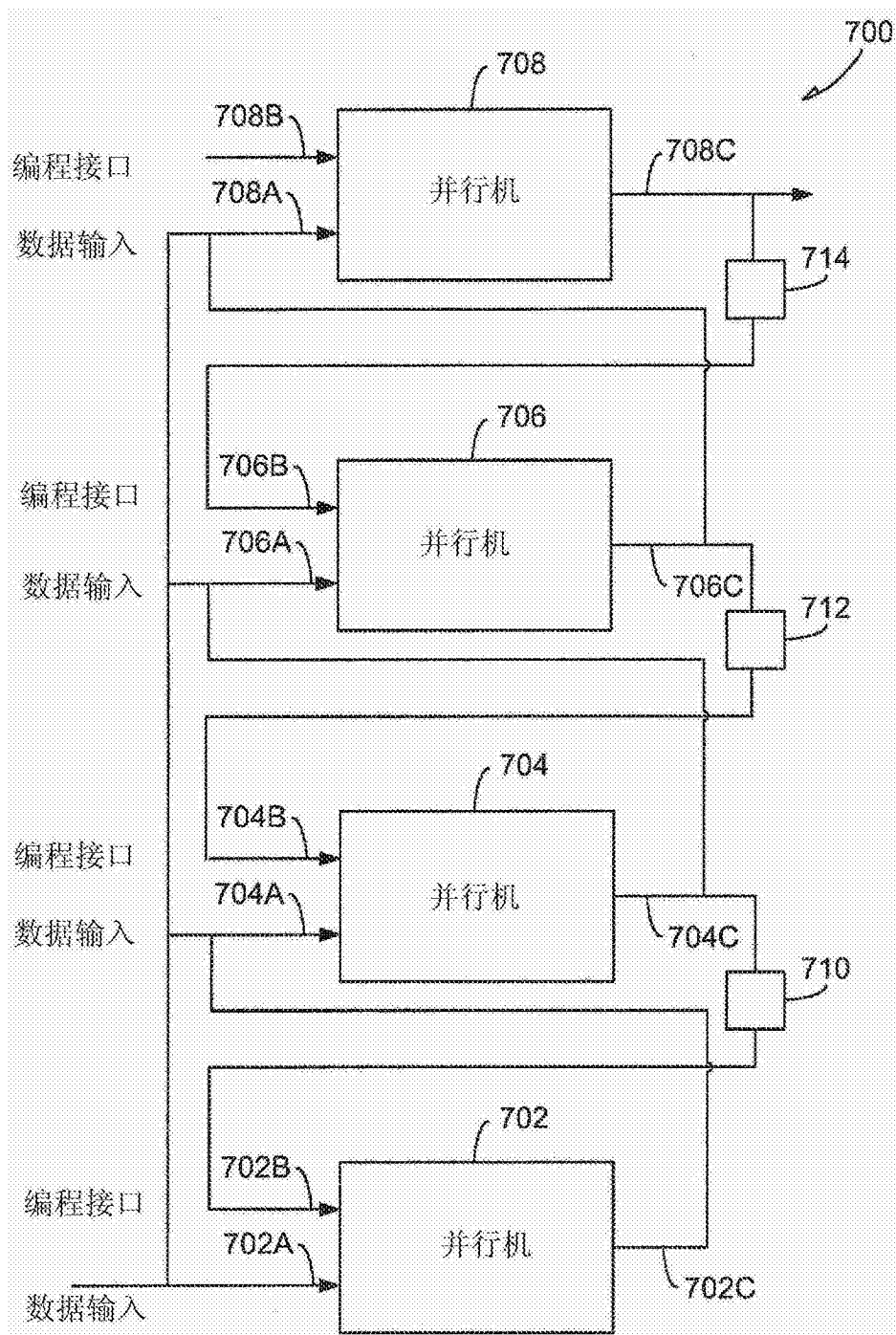


图 7

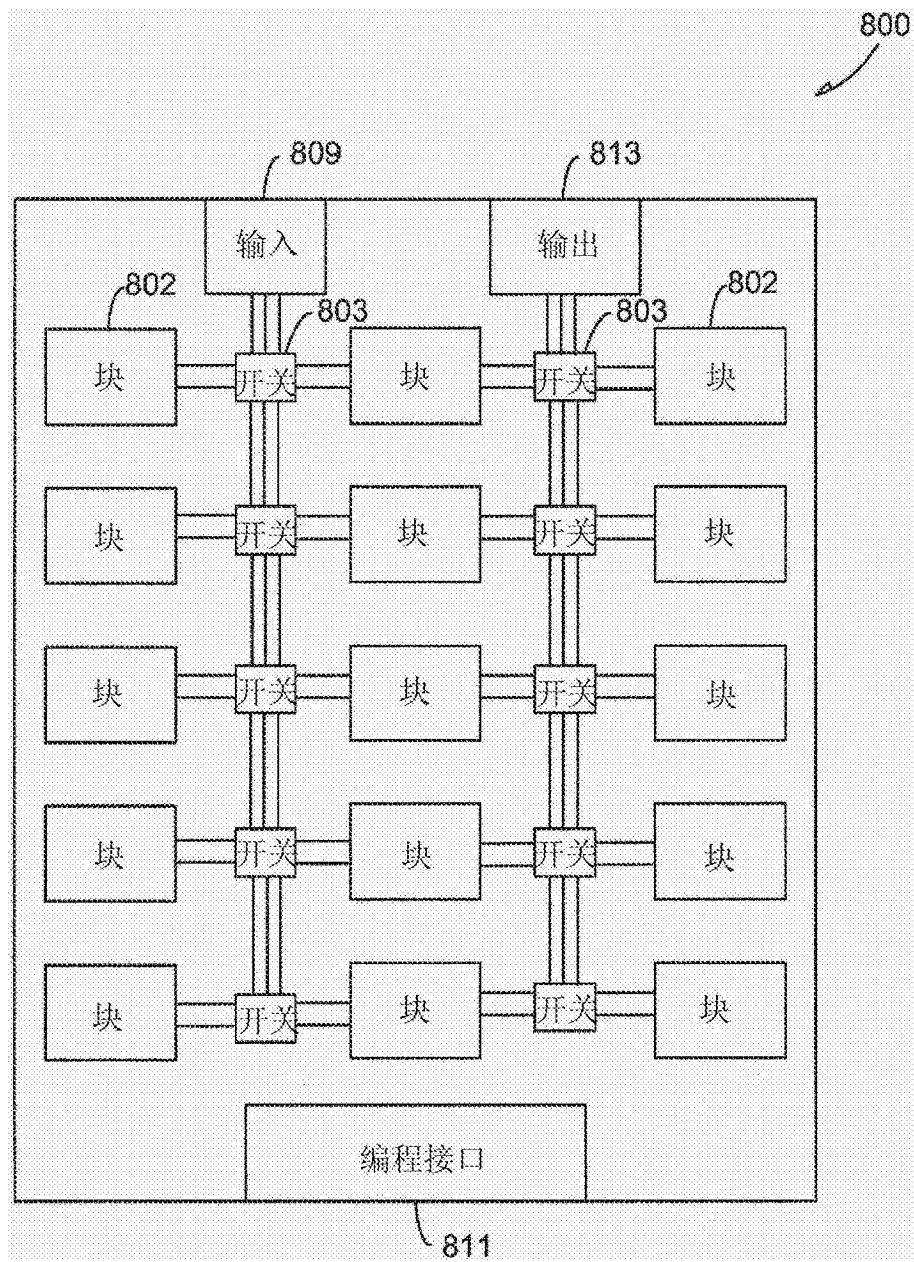


图 8

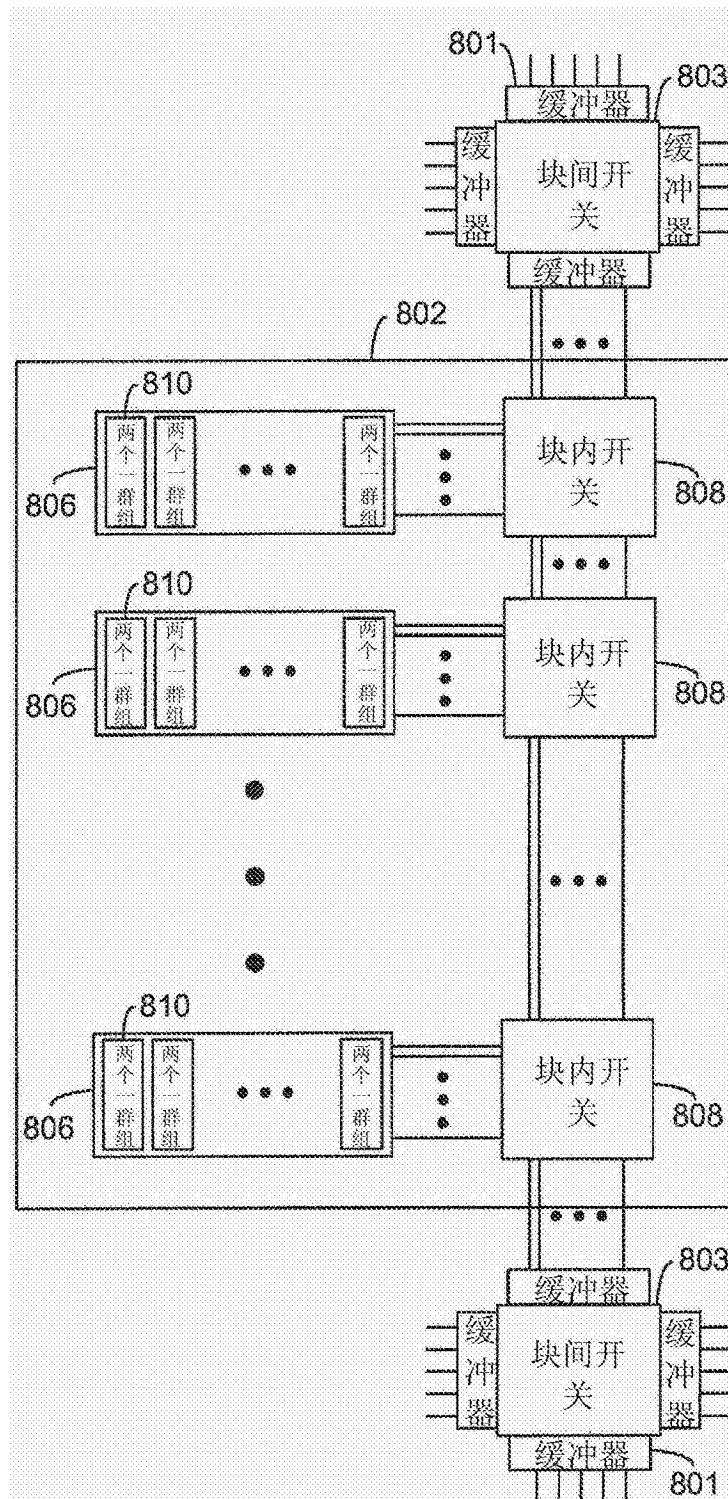


图 9

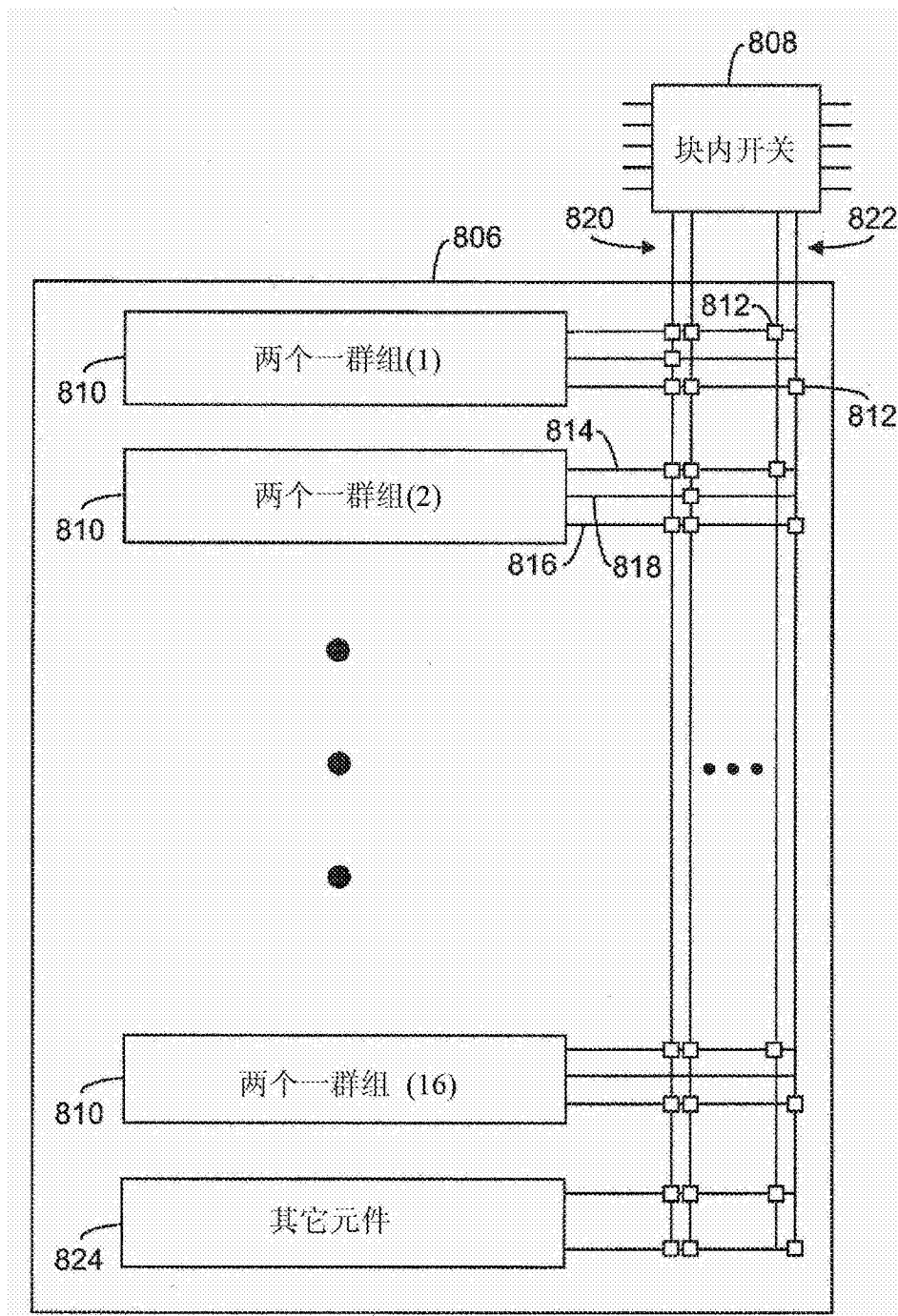


图 10

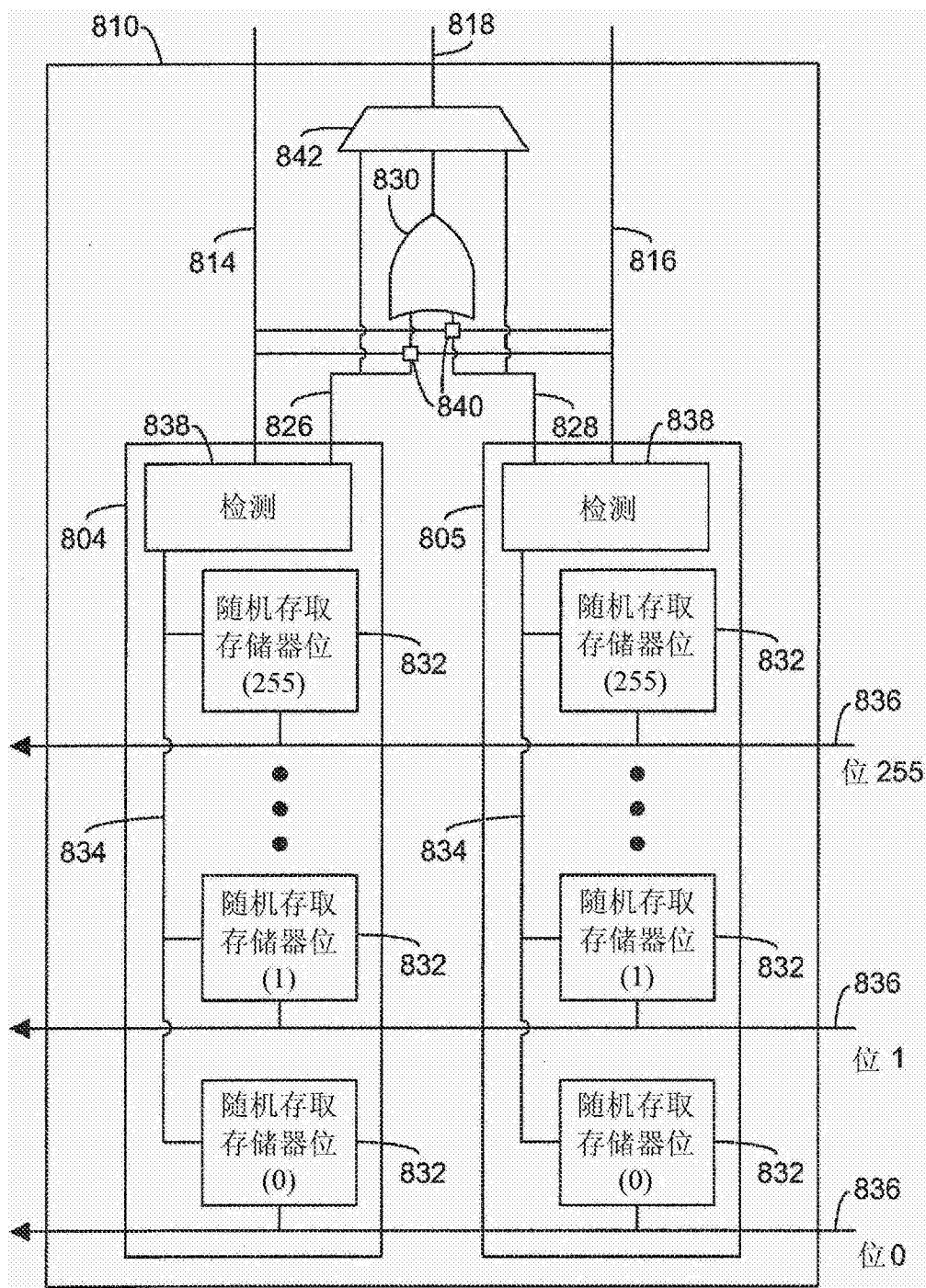
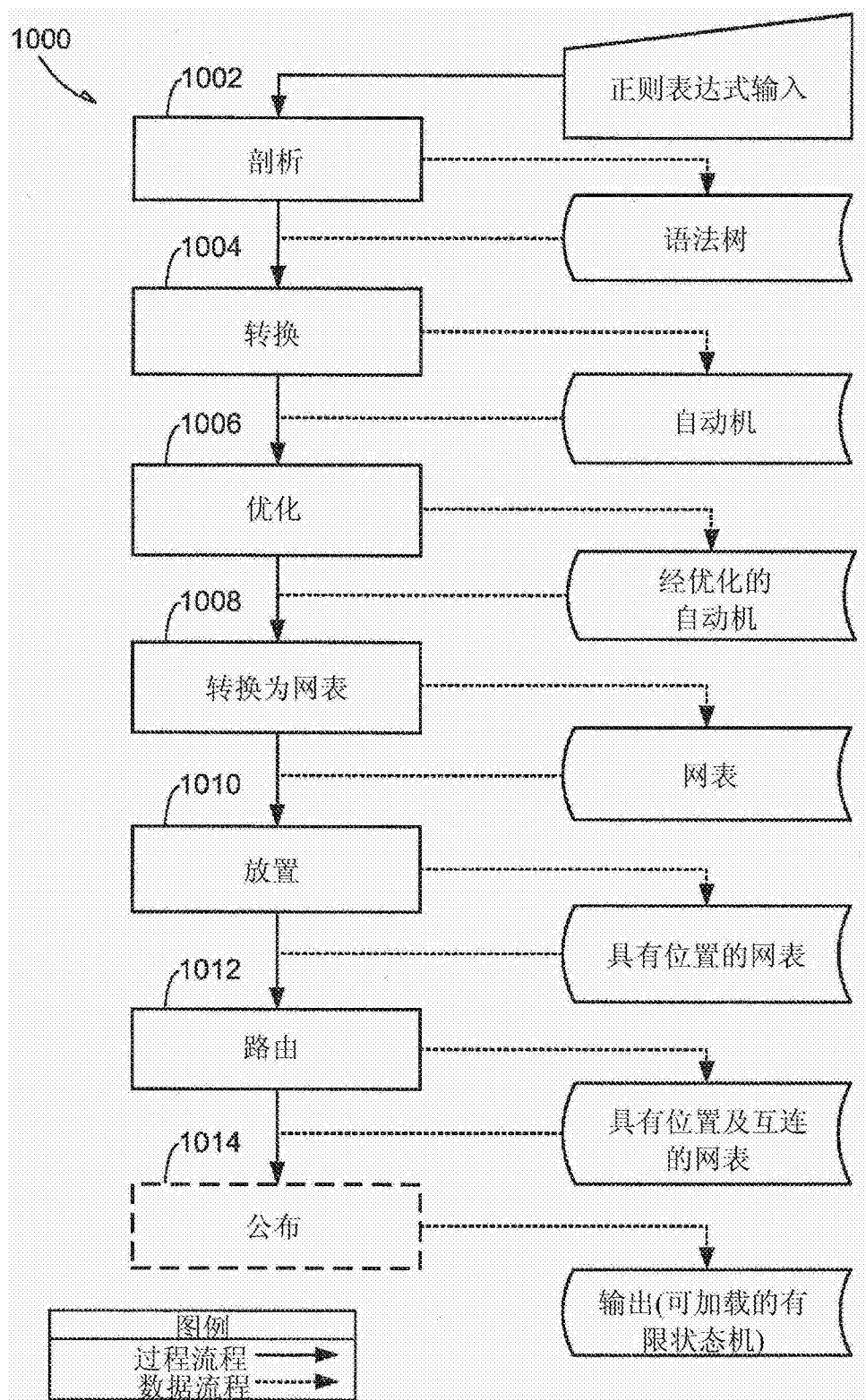


图 11



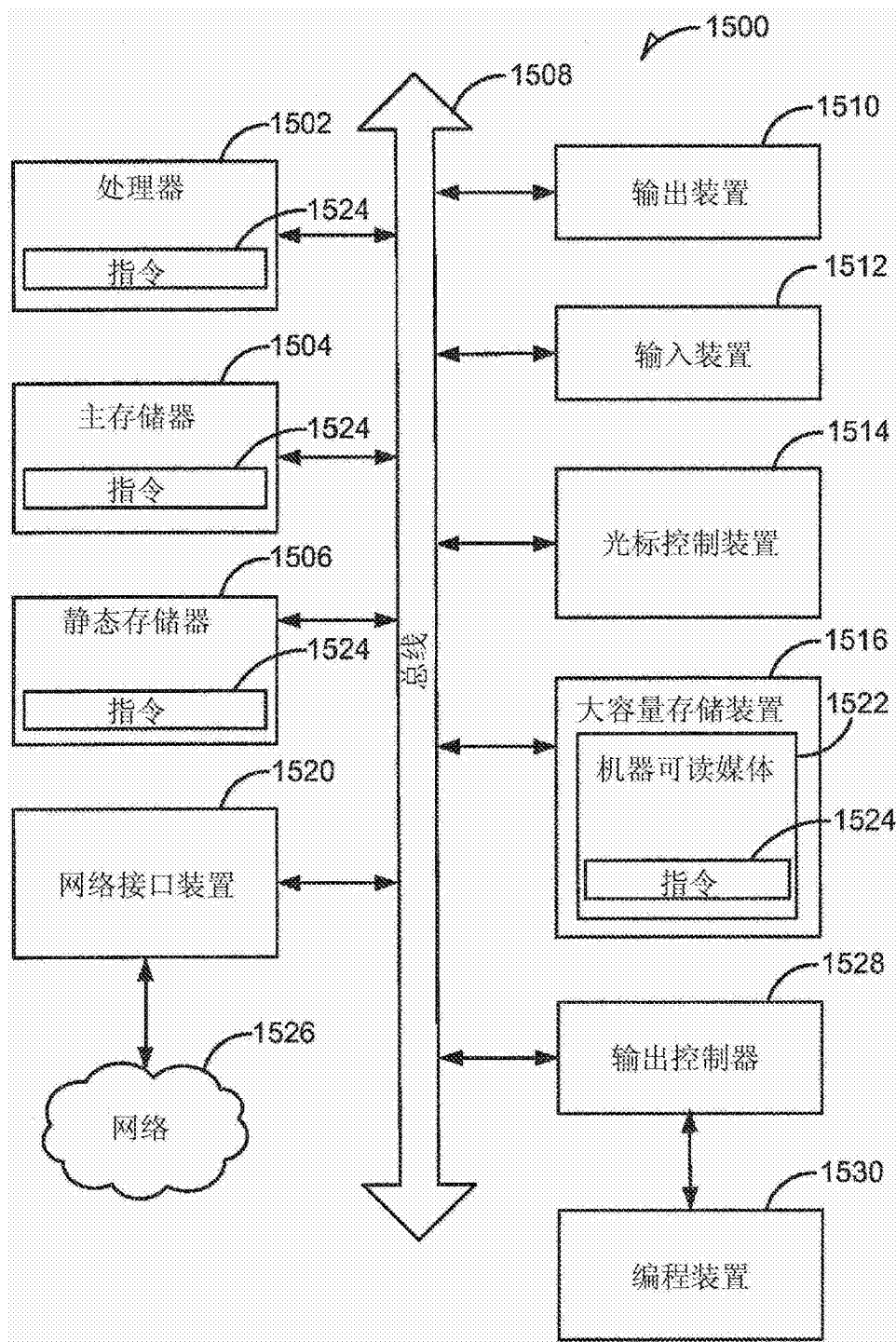


图 13