(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105912992 B
(45)授权公告日 2019.12.10

(21)申请号 201610206769.8
(22)申请日 2011.06.09
(65)同一申请的已公布的文献号
申请公布号 CN 105912992 A
(43)申请公布日 2016.08.31
(30)优先权数据
61/353,546 2010.06.10 US
12/943,551 2010.11.10 US
(62)分案原申请数据
201180034987.3 2011.06.09
(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 路勇
(51)Int.Cl.
G06K 9/00(2006.01)
审查员 乔晋

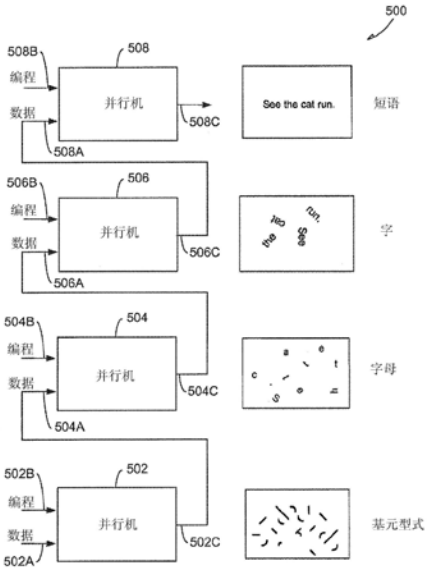
权利要求书4页 说明书16页 附图13页

(54)发明名称

使用层次结构分析数据

(57)摘要

本申请涉及使用层次结构分析数据。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。



CN 105912992 B

1.一种用于使用硬件处理元件的可配置状态来分析数据型式以执行操作的方法，其中所述硬件处理元件包括在装置中，所述装置可配置以实施状态机元件的布置以基于由可编程开关控制的所述状态机元件之间的互连，经由所述状态机元件实施第一并行机和第二并行机，且所述操作包括：

使用所述第一并行机搜索第一型式，以确定第一数据集是否包括所述第一型式，其中所述第一并行机是在具有所述硬件处理元件的第一配置的所述装置中实施，其中所述第一配置经由第一组可编程开关从所述硬件处理元件的第一互连实施，且其中所述第一配置将所述硬件处理元件布置为第一有限状态机的相应状态；

从所述第一并行机输出所述第一数据集是否包括所述第一型式的指示；

使用所述第二并行机搜索第二型式，以确定第二数据集是否包括所述第二型式，所述确定基于所述第一数据集是否包括所述第一型式的所述指示，其中所述第二并行机是在具有所述硬件处理元件的第二配置的所述装置中实施，其中所述第一数据集是否包括所述第一型式的所述指示至少部分地确定所述硬件处理元件的所述第二配置，其中所述第二配置经由第二组可编程开关从所述硬件处理元件的第二互连实施，且其中所述第二配置将所述硬件处理元件布置为第二有限状态机的相应状态；以及

从所述第二并行机输出所述第二数据集是否包括所述第二型式的指示；

基于所述第二数据集是否包括所述第二型式的所述指示来编程另一并行机的硬件处理元件，其中所述另一并行机用所述另一并行机的硬件处理元件的另一配置来实施，其中所述第二数据集是否包括所述第二型式的所述指示至少部分地确定所述另一并行机的所述硬件处理元件的所述配置，其中编程所述另一并行机的所述硬件处理元件经由第三组可编程开关创建第三配置，且其中所述第三配置将所述另一并行机的所述硬件处理元件布置成用于型式匹配操作的第三有限状态机的相应状态。

2.根据权利要求1所述的方法，其中所述第一数据集是否包括所述第一型式的所述指示是在所述第二数据集内提供。

3.根据如权利要求1所述的方法，其中所述另一并行机是所述第一并行机，且其中所述编程基于所述第二数据集是否包括所述第二型式的所述指示来配置所述第一并行机以用于型式匹配操作。

4.根据权利要求1所述的方法，进一步包括从所述第一并行机输出所述第二数据集以由所述第二并行机使用，所述第二数据集基于实施所述第一并行机的可编程硬件处理元件对所述第一数据集的反应。

5.根据权利要求4所述的方法，其中所述第一数据集是否包括所述第一型式的所述指示是在所述第二数据集内提供，其中所述第二数据集包括字，所述字具有指示与多个型式的匹配及不匹配的多个位，且其中所述多个型式包括所述第一型式。

6.根据权利要求4所述的方法，其中所述第二数据集包括状态向量，所述状态向量对应于在型式匹配之后实施所述第一并行机的可编程硬件处理元件的状态。

7.根据权利要求1所述的方法，其中使用所述第一并行机确定所述第一数据集是否包括所述第一型式包括确定所述第一数据集是否包括所关注序列。

8.根据权利要求1所述的方法，其中使用所述第一并行机确定所述第一数据集是否包括所述第一型式包括确定是否在所述第一数据集中辨识出字符序列。

9.根据权利要求1所述的方法，进一步包括使用所述第二并行机器确定所述第一数据集是否包括所述第二型式。

10.根据权利要求1所述的方法，进一步包括基于由所述第二并行机在所述第二数据集中辨识出的所述第二型式，重新配置用于所述型式匹配操作的所述第一并行机。

11.根据权利要求10所述的方法，其中重新配置用于所述型式匹配操作的所述第一并行机包括：基于在所述第二数据集中辨识出的所述第二型式，编译所述第一并行机的程序以执行所述型式匹配操作，其中所述程序定义所述第二互连。

12.根据权利要求1所述的方法，其中所述另一并行机是第三并行机，其中所述第三并行机用所述硬件处理元件的第三可配置状态实施，且所述方法还包括：

使用所述第三并行机确定第三数据集是否包括第三型式；

从所述第三并行机输出在所述第三数据集中是否辨识出所述第三型式的指示；以及

基于在所述第三数据集中是否辨识出所述第三型式的所述指示，配置所述第一并行机或所述第二并行机中的至少一者。

13.一种用于处理数据的设备，包括：

可编程开关的层次结构；

包括状态机元件的硬件处理元件，其中所述硬件处理元件的给定状态机元件是表示给定有限状态机中的单个状态的物理组件，且其中所述硬件处理元件经由可编程开关耦合；

从第一多个硬件处理元件的第一配置实施的第一并行机的电路，其中所述第一配置从经由第一组可编程开关的所述硬件处理元件的第一互连以及所述状态机元件到第一有限状态机中的相应状态的第一映射提供，且其中所述第一并行机经配置以：

搜索以确定第一数据集是否包括第一型式；以及

产生在所述第一数据集中是否辨识出所述第一型式的指示；以及

从第二多个硬件处理元件的第二配置实施的第二并行机的电路，所述第二并行机响应于在所述第一数据集中是否辨识出所述第一型式的所述指示来调适所述第二多个硬件处理元件的所述第二配置中的操作，其中所述第二配置从经由第二组可编程开关的所述硬件处理元件的第二互连以及所述状态机元件到第二有限状态机中的相应状态的第二映射提供，且其中所述第二并行机经配置以：

搜索第二型式以确定第二数据集是否包括所述第二型式；以及

产生在所述第二数据集中是否辨识出所述第二型式的指示；

基于在所述第二数据集中是否辨识出所述第二型式的所述指示来编程另一并行机的硬件处理元件的电路，其中所述另一并行机用所述另一并行机的硬件处理元件的另一配置来实施，且其中在所述第二数据集中是否辨识出所述第二型式的所述指示至少部分地确定所述硬件处理元件的所述另一配置，其中使用所述另一并行机的所述硬件处理元件上的程序经由第三组可编程开关创建第三配置，且其中所述第三配置将所述另一并行机的所述硬件处理元件布置成用于型式匹配操作的第三有限状态机的相应状态。

14.根据权利要求13所述的设备，进一步由第三多个硬件处理元件的所述第三配置实施的第三并行机的电路，所述第三并行机基于在所述第二数据集中是否辨识出所述第二型式的所述指示执行操作。

15.根据权利要求13所述的设备，其中所述第二并行机进一步经配置以：

在由所述第一并行机产生的数据流中接收所述第二数据集;以及

辨识所述数据流中的所述第二型式。

16.根据权利要求13所述的设备,其中所述第一并行机包括接口以基于在所述第二数据集中是否辨识出所述第二型式的所述指示来接收所述第一并行机的重新配置以用于后续操作。

17.根据权利要求16所述的设备,进一步包括耦合到所述第二并行机的处理装置,所述处理装置经配置以:

分析在所述第二数据集中是否辨识出所述第二型式的所述指示;以及

根据在所述第二数据集中是否辨识出所述第二型式,编译所述第一并行机的更新程序,其中所述更新程序通过定义所述第二互连和所述第二映射来实施所述重新配置。

18.一种非暂时性计算机可读存储介质,所述非暂时性计算机可读存储介质包括用于使用硬件处理元件的可配置状态来分析数据型式的指令,其中所述硬件处理元件包括在装置中,所述装置可配置以实施状态机元件的布置,其中所述指令在执行时实施第一并行机和第二并行机,所述第一并行机和所述第二并行机经由所述状态机元件和由可编程开关控制的所述状态机元件之间的互连来实施,所述指令在执行时具有以下操作:

配置第一并行机,其中所述第一并行机使用所述硬件处理元件的第一配置来实施,其中所述第一配置从经由第一组可编程开关的所述硬件处理元件的第一互连以及所述状态机元件到第一有限状态机中的相应状态的第一映射提供,且其中所述第一并行机包括第一数据输入端口和第一输出端口,且所述第一并行机由所述指令配置以:

在所述第一数据输入端口处接收的第一数据中搜索第一型式;以及

在所述第一输出端口处输出第二数据,所述第二数据提供在所述第一数据中是否辨识出所述第一型式的指示;以及

配置第二并行机,其中所述第二并行机包括第二数据输入端口和第二输出端口,其中所述第二并行机使用所述硬件处理元件的第二配置来实施,其中所述第二配置从经由第二组可编程开关的所述硬件处理元件的第二互连以及所述状态机元件到第二有限状态机中的相应状态的第二映射提供,且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,且所述第二并行机由所述指令配置以:

在所述第二数据输入端口处从所述第一并行机接收的所述第二数据中搜索第二型式,且其中在所述第一数据中是否辨识出所述第一型式的所述指示至少部分地确定所述硬件处理元件的所述第二配置;以及

在所述第二输出端口输出第三数据,第三数据提供在所述第二数据中是否辨识出所述第二型式的指示;

基于所述第二数据集是否包括所述第二型式的所述指示来产生程序以配置另一并行机的硬件处理元件,其中所述另一并行机用所述另一并行机的硬件处理元件的另一配置来实施,其中所述第二数据集是否包括所述第二型式的所述指示至少部分地确定所述硬件处理元件的所述另一配置;

其中使用所述另一并行机的所述硬件处理元件上的程序经由第三组可编程开关创建第三配置,且其中所述第三配置将所述另一并行机的所述硬件处理元件布置成用于型式匹配操作的第三有限状态机的相应状态。

19.根据权利要求18所述的非暂时性计算机可读存储介质，其中所述第一并行机包括能够基于所述第三数据接收程序的接口，所述程序能够基于在所述第二数据中是否辨识出所述第二型式的指示来重新配置所述第一并行机。

20.根据权利要求18所述的非暂时性计算机可读存储介质，其中所述第二并行机进一步由所述指令配置以：

在由所述第一并行机产生的数据流中接收所述第二数据；且

辨识所述数据流中的所述第二型式。

21.一种用于分析数据的方法，包括：

使用用第一多个硬件元件实施的第一并行机分析第一数据，所述分析包括确定所述第一数据是否匹配第一型式；

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；

使用用第二多个硬件元件实施的第二并行机分析所述第二数据，所述分析包括辨识所述第二数据匹配第二型式；及

通过输出第三数据，从所述第二并行机输出分析所述第二数据的结果，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型式的指示：

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型式；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型式的指示。

22.根据权利要求21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。

使用层次结构分析数据

[0001] 本申请为申请日为2011年6月9日、申请号为201180034987.3、发明名称为“使用层次结构分析数据”的原中国发明专利申请的分案申请。

[0002] 相关申请案的交叉参考

[0003] 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年月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

[0004] 本发明涉及使用层次结构分析数据。

背景技术

[0005] 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

[0006] 本申请涉及一种用于分析数据的方法,其包含:使用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;使用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。

附图说明

[0007] 图1图解说明根据本发明的各种实施例的并行机的实例。

[0008] 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。

[0009] 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。

[0010] 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。

[0011] 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。

[0012] 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。

[0013] 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。

[0014] 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。

[0015] 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。

[0016] 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。

[0017] 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。

[0018] 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。

[0019] 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

[0020] 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。

[0021] 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)。

[0022] 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。

[0023] 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。

[0024] 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。

[0025] 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)。

[0026] 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。

[0027] 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。

[0028] 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。

[0029] 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。

[0030] 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。

[0031] 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。

[0032] 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。

[0033] 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。

[0034] 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。

[0035] 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。

[0036] 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。

[0037] 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型式可由所述终端节点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。

[0038] 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。

[0039] 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。

[0040] 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。

[0041] 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。

[0042] 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。

[0043] 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。

[0044] 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。

[0045] 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层级，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。

[0046] 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。此些型式(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。

[0047] 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。

[0048] 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,较高层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层级中所识别的型式的分量。

[0049] 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)。

[0050] 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。

[0051] 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层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。

[0052] 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。

[0053] 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。

[0054] 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。

[0055] 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)。

[0056] 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。

[0057] 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。

[0058] 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。

[0059] 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。

[0060] 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可从来自较低层级与原始数据的型式的组合识别型式。

[0061] 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。

[0062] 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层级对象。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,在较低层级处越快地接收反馈,所述较低层级便可越快地调适且分析越高效。

[0063] 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。

[0064] 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。

[0065] 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。

[0066] 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。

[0067] 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。

[0068] 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。

[0069] 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。

[0070] 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。

[0071] 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。

[0072] 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。

[0073] 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/或传送用以重新编程FSM引擎800或重新编程另一FSM引擎800的编程信息。

[0074] 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。

[0075] 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ACSII字符转换成256个位中的1。

[0076] 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。

[0077] 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。

[0078] 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SME 804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SME 804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SME 805或供从FSM引擎800输出。

[0079] 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)。

[0080] 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。

[0081] 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。

[0082] 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。

[0083] 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。

[0084] 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。

[0085] 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。

[0086] 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，δ(p,α)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→q在同一符号上反转q→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。

[0087] 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。

[0088] 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。

[0089] 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。

[0090] 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。

[0091] 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。

[0092] 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。

[0093] 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。

[0094] 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。此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。

[0095] 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。

[0096] 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。

[0097] 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。

[0098] 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对等(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。

[0099] 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。

[0100] 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。

[0101] 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。

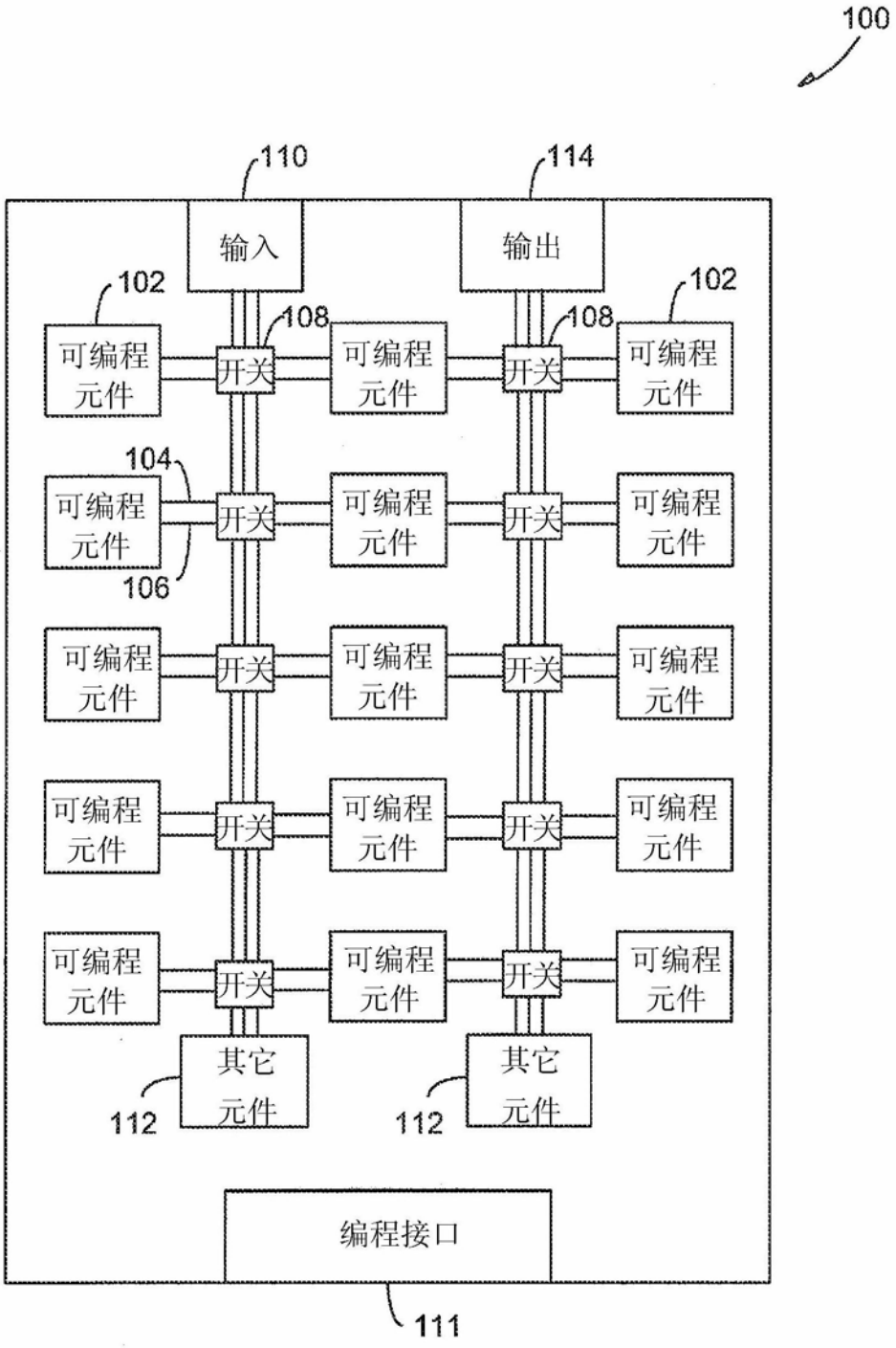

100
110
114
输入
输出
102
102
108
108
可编程元件
开关
可编程元件
开关
可编程元件
104
106
其它元件
112
其它元件
112
编程接口
111

图1

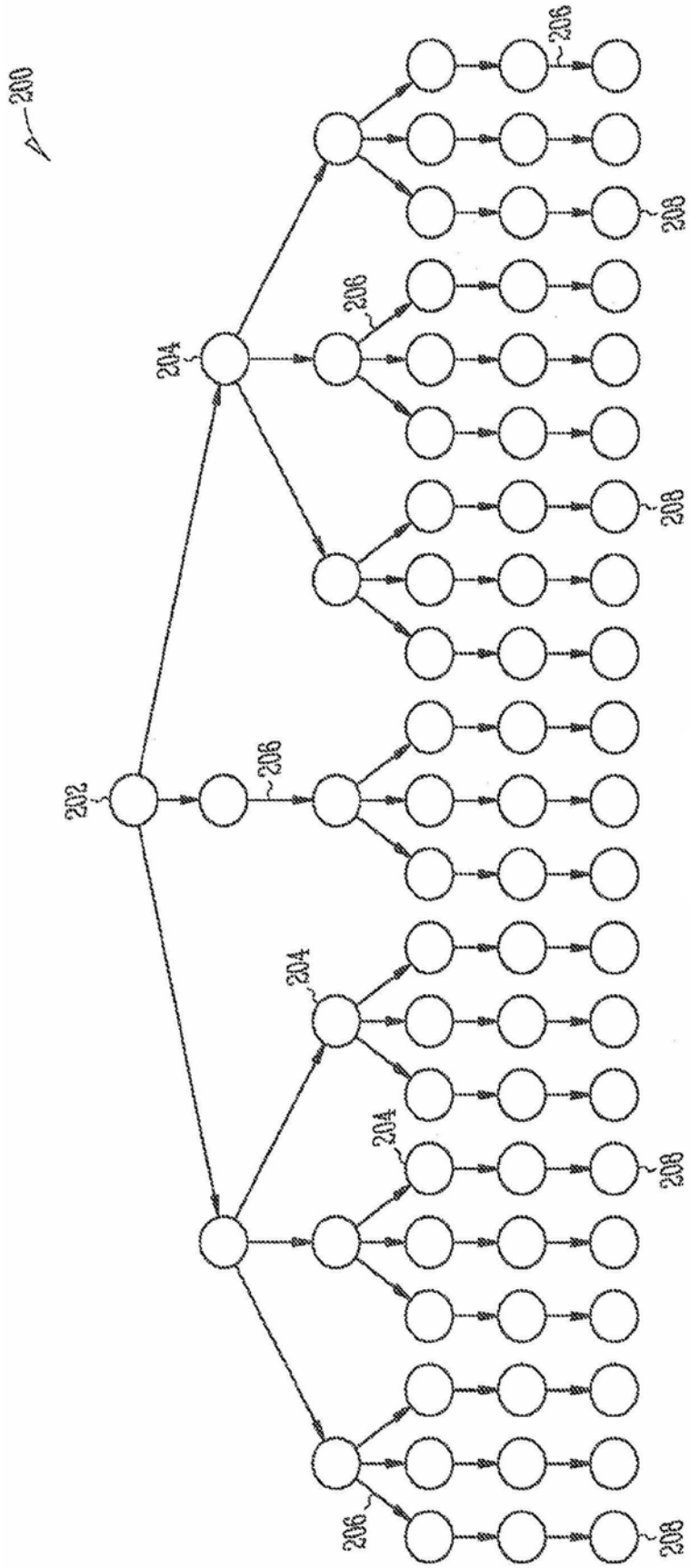


图2

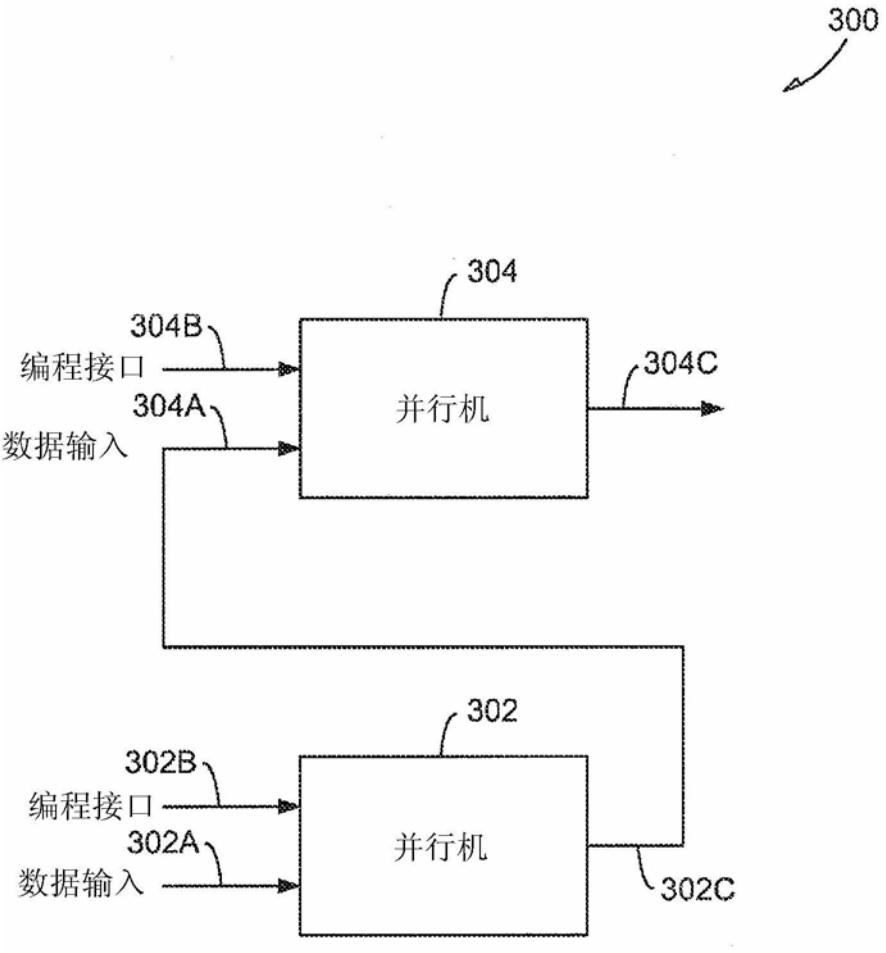

300
304
304B
编程接口
304A
数据输入
并行机
304C
302
302B
编程接口
302A
数据输入
并行机
302C

图3

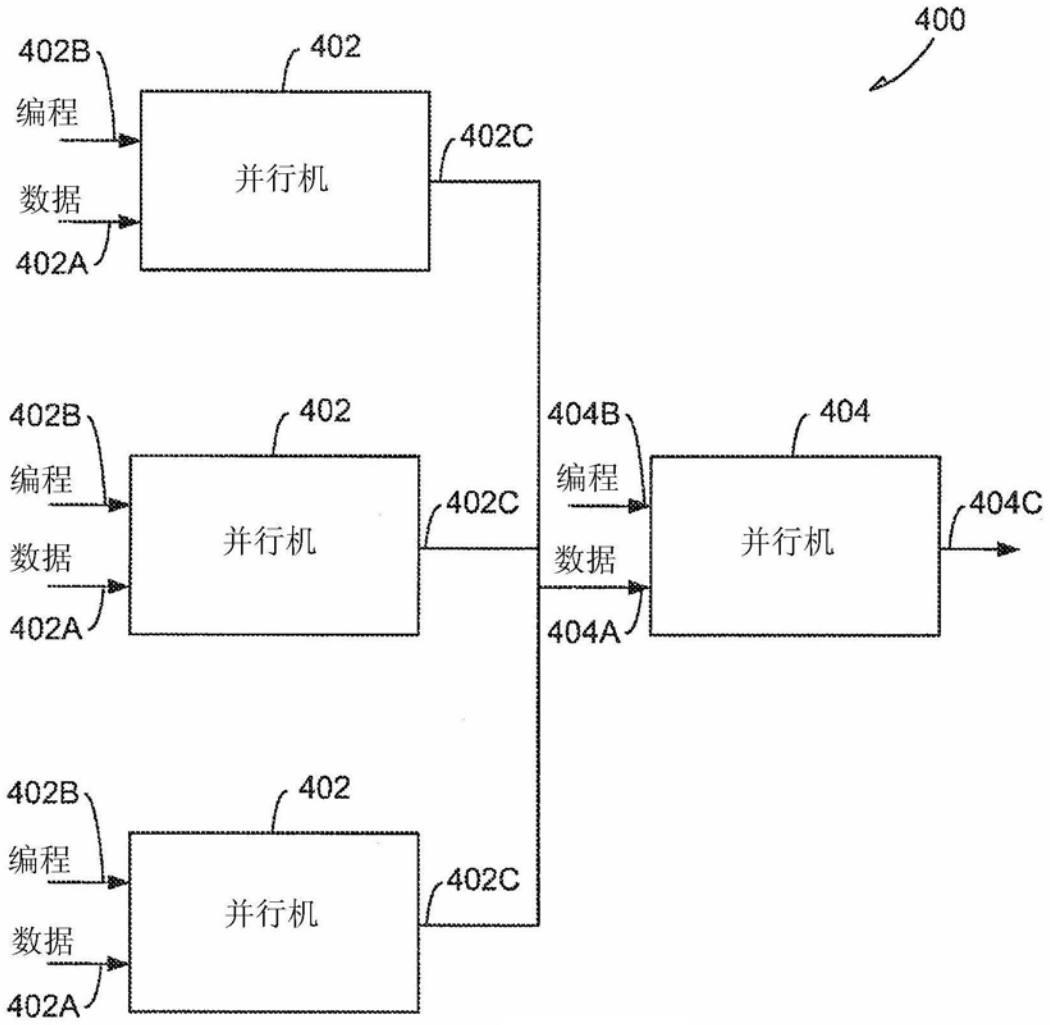

400
402B
402
编程
并行机
402C
数据
402A
402B
402
编程
并行机
402C
数据
402A
404B
404
编程
并行机
404C
数据
404A
402B
402
编程
并行机
402C
数据
402A

图4

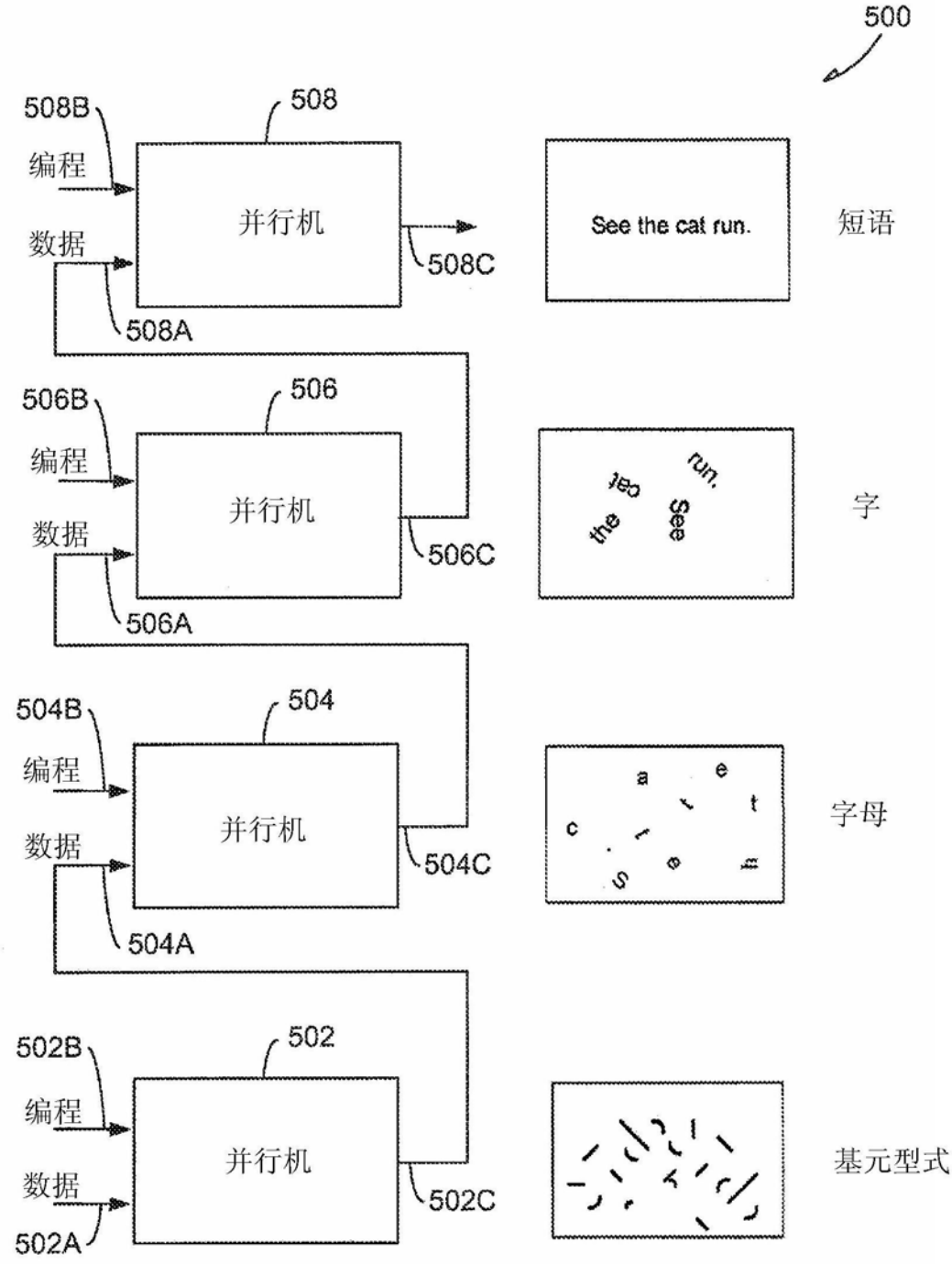

500
508B
编程
数据
508
并行机
508C
508A
See the cat run.
短语
506B
编程
数据
506
并行机
506C
506A
run.
cat
the
See
字
504B
编程
数据
504
并行机
504C
504A
字母
502B
编程
数据
502
并行机
502C
502A
基元型式

图5

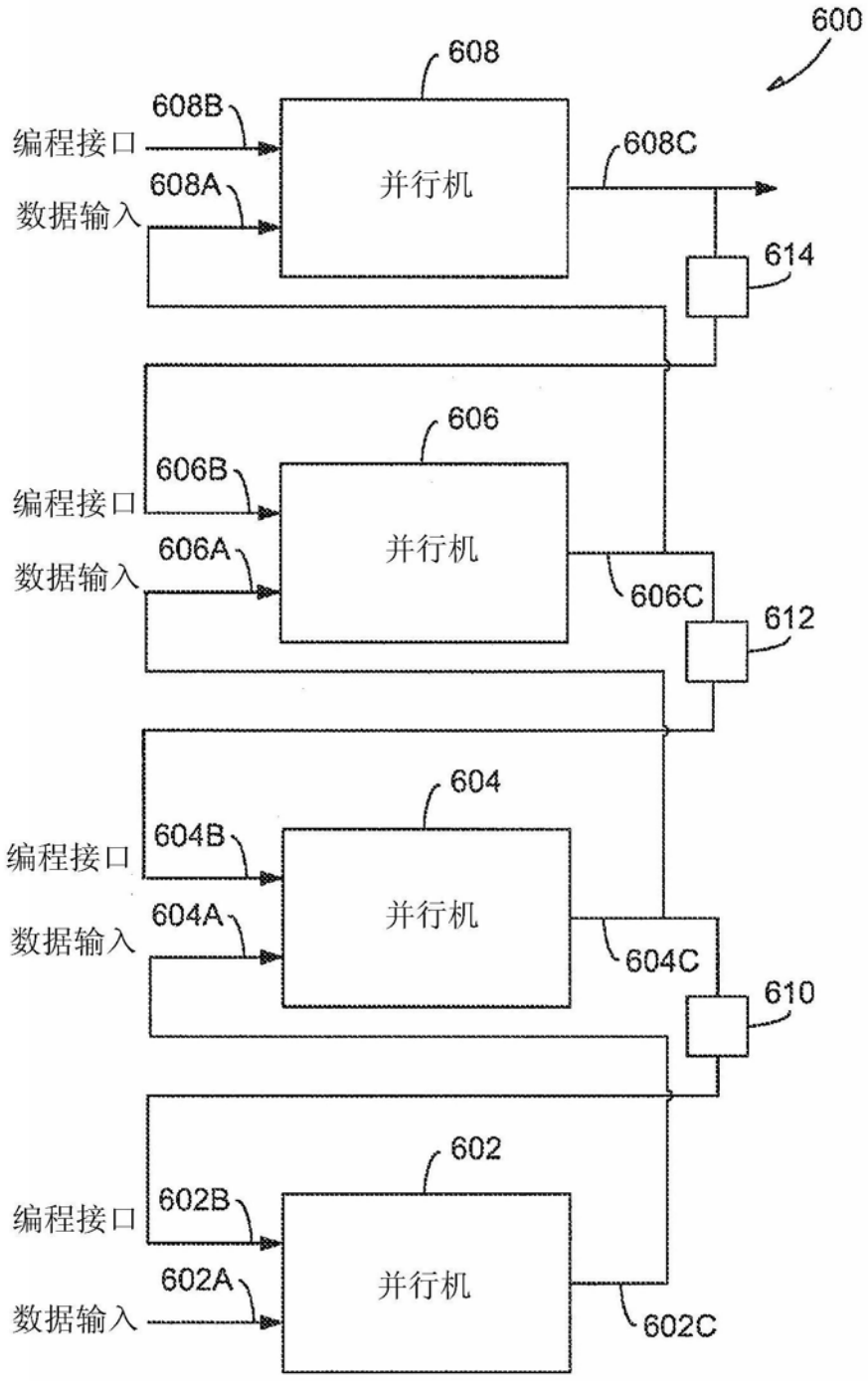

600
608
608B
编程接口
608A
数据输入
并行机
608C
614
606
606B
编程接口
606A
数据输入
并行机
606C
612
604
604B
编程接口
604A
数据输入
并行机
604C
610
602
602B
编程接口
602A
数据输入
并行机
602C

图6

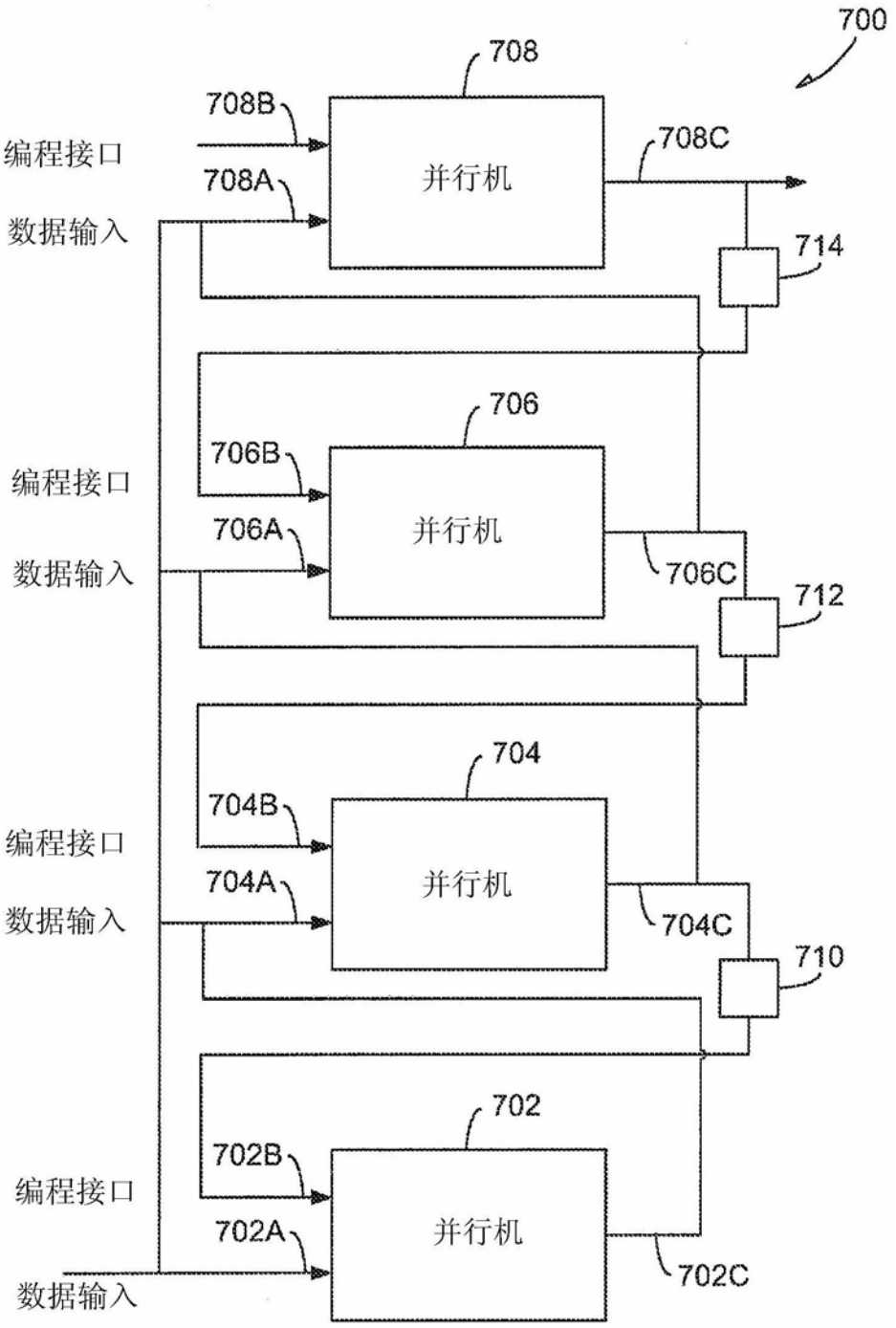

700
708
708B
编程接口
708A
数据输入
并行机
708C
714
706
706B
编程接口
706A
数据输入
并行机
706C
712
704
704B
编程接口
704A
数据输入
并行机
704C
710
702
702B
编程接口
702A
数据输入
并行机
702C

图7

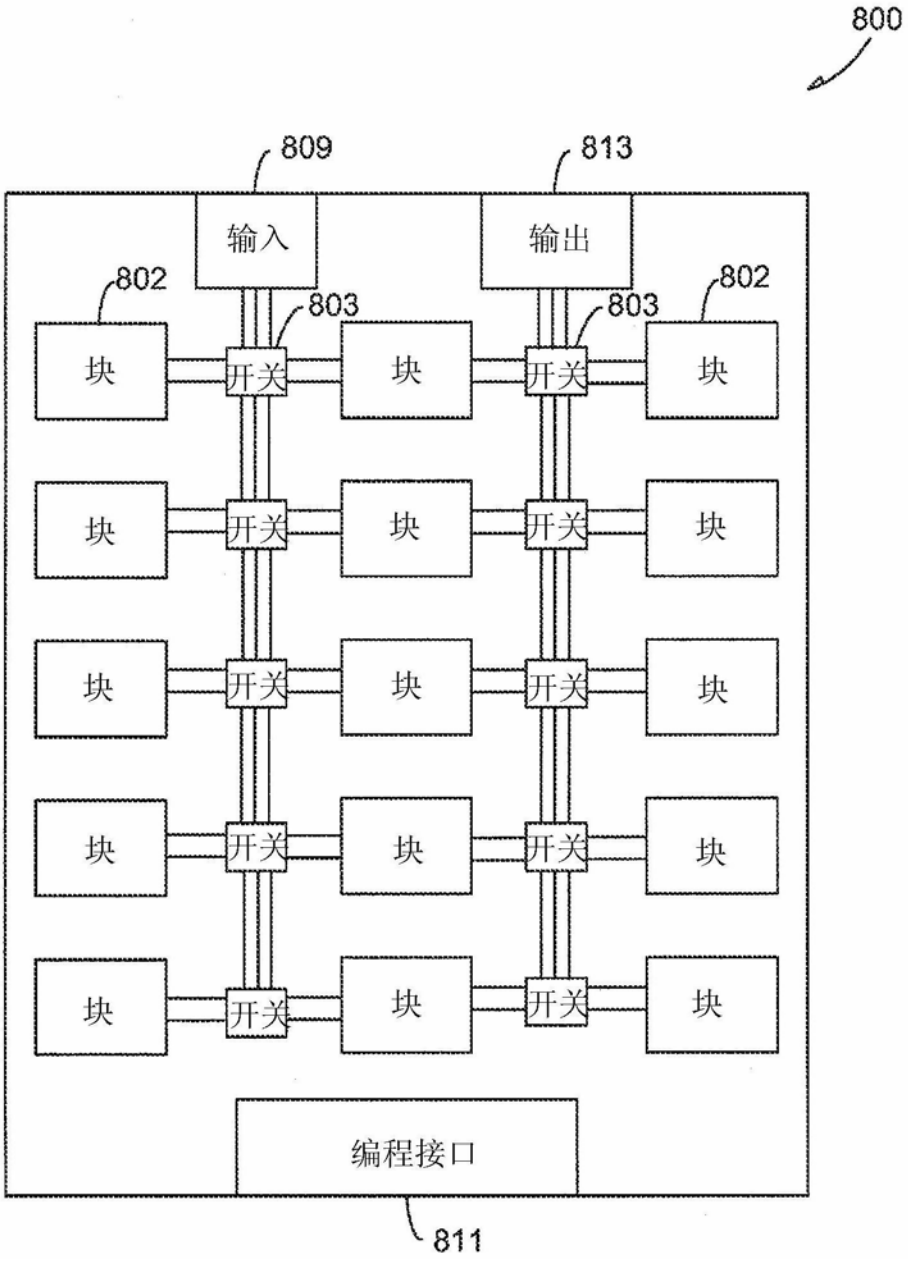


图8

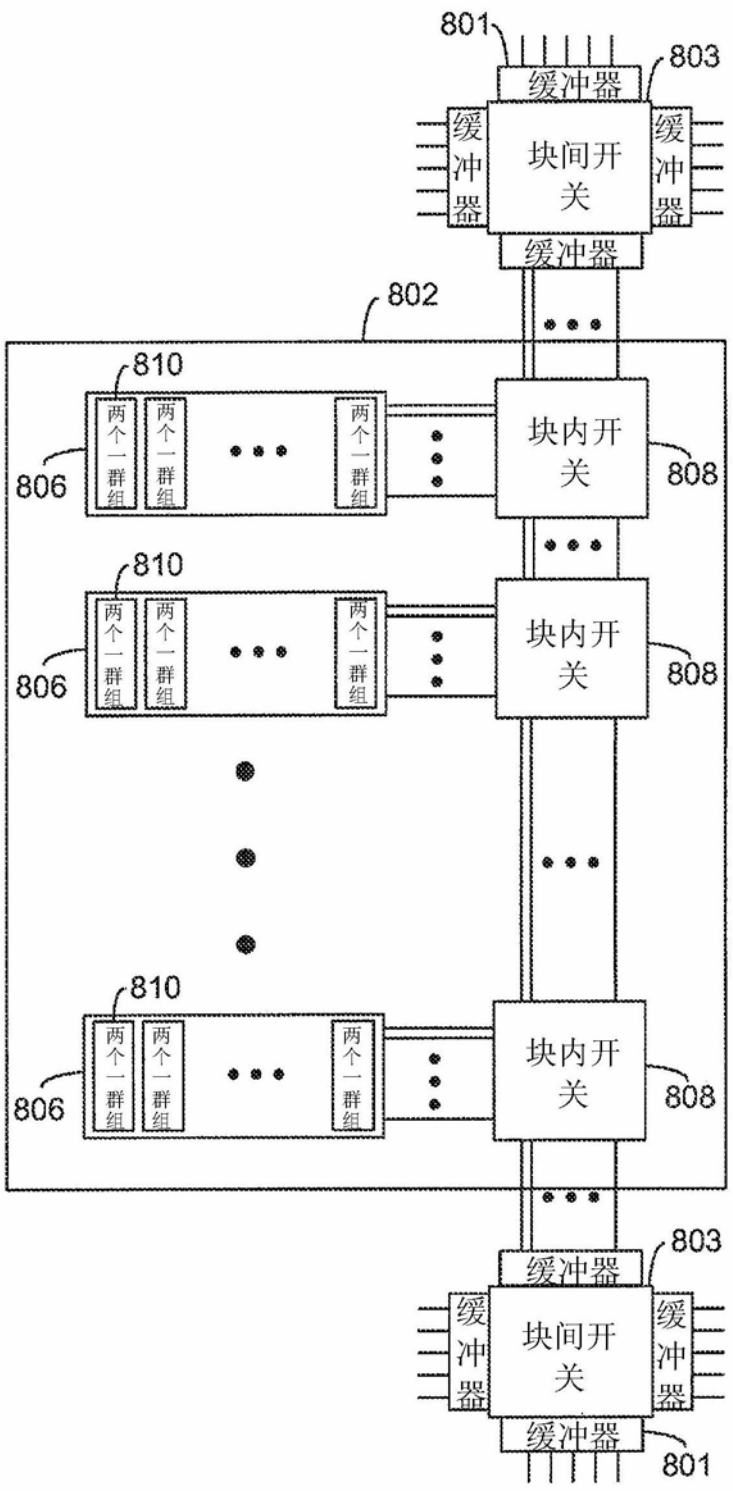

801
803
缓冲器
缓冲器
缓冲器
块间开关
缓冲器
802
810
两个一群组
两个一群组
两个一群组
806
块内开关
808
缓冲器
缓冲器
缓冲器
块间开关
缓冲器
801

图9

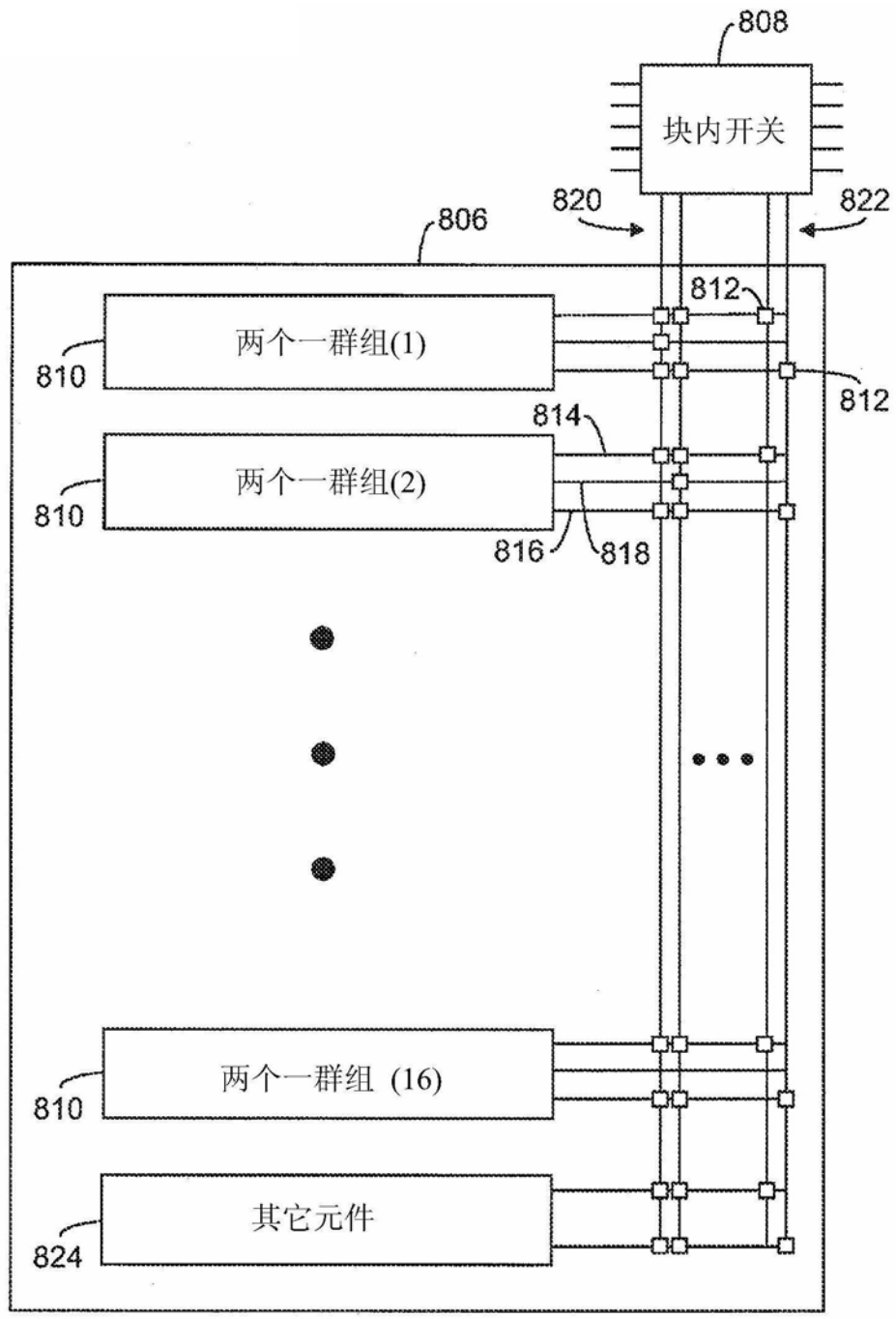

808
块内开关
820
822
806
812
两个一群组(1)
810
812
814
两个一群组(2)
810
816
818
两个一群组 (16)
810
其它元件
824

图10

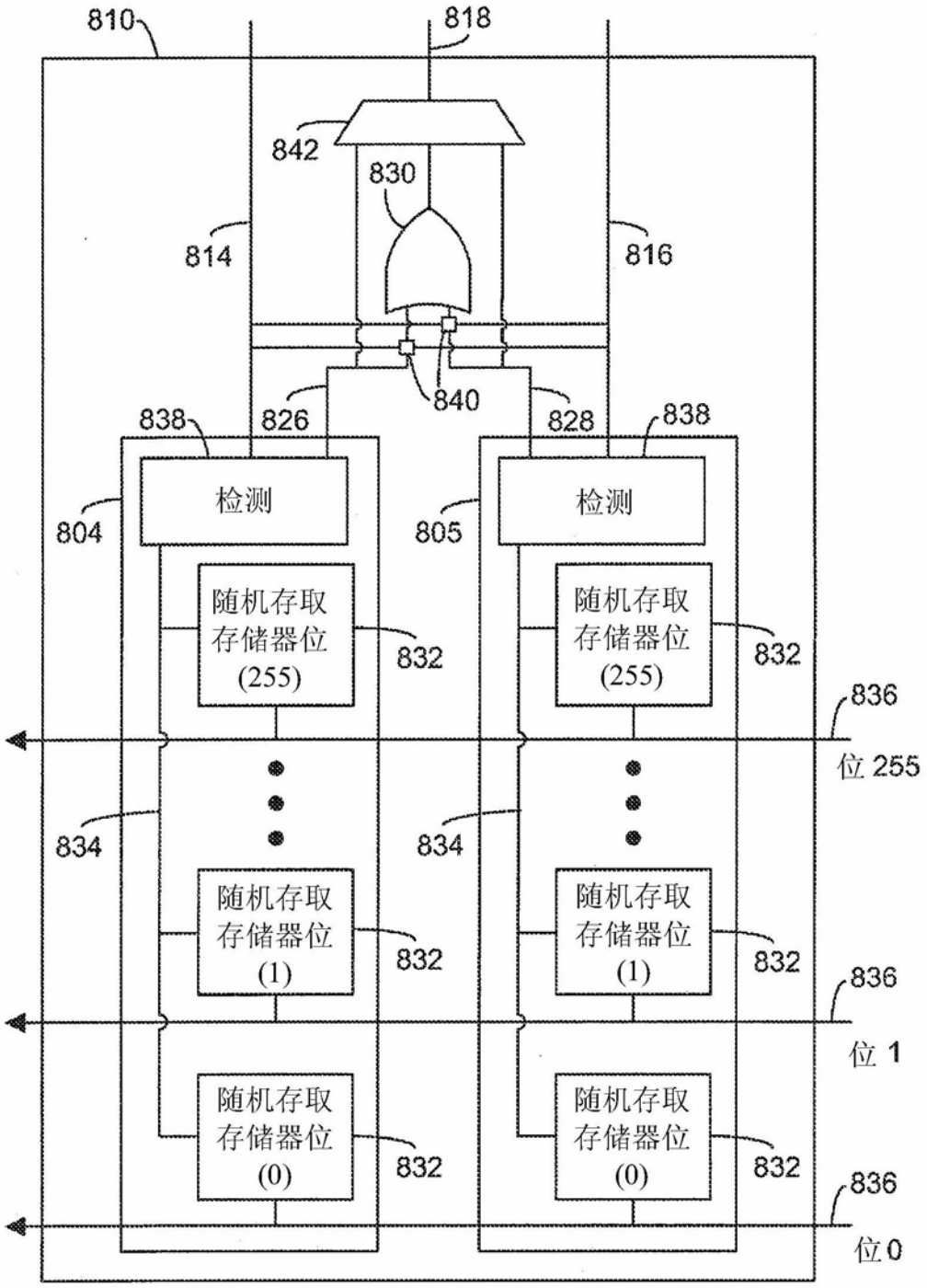

810
818
842
830
814
816
840
826
828
838
838
检测
检测
804
805
随机存取存储器位(255)
随机存取存储器位(255)
832
832
836
位 255
834
834
随机存取存储器位(1)
随机存取存储器位(1)
832
832
836
位 1
随机存取存储器位(0)
随机存取存储器位(0)
832
832
836
位 0

图11

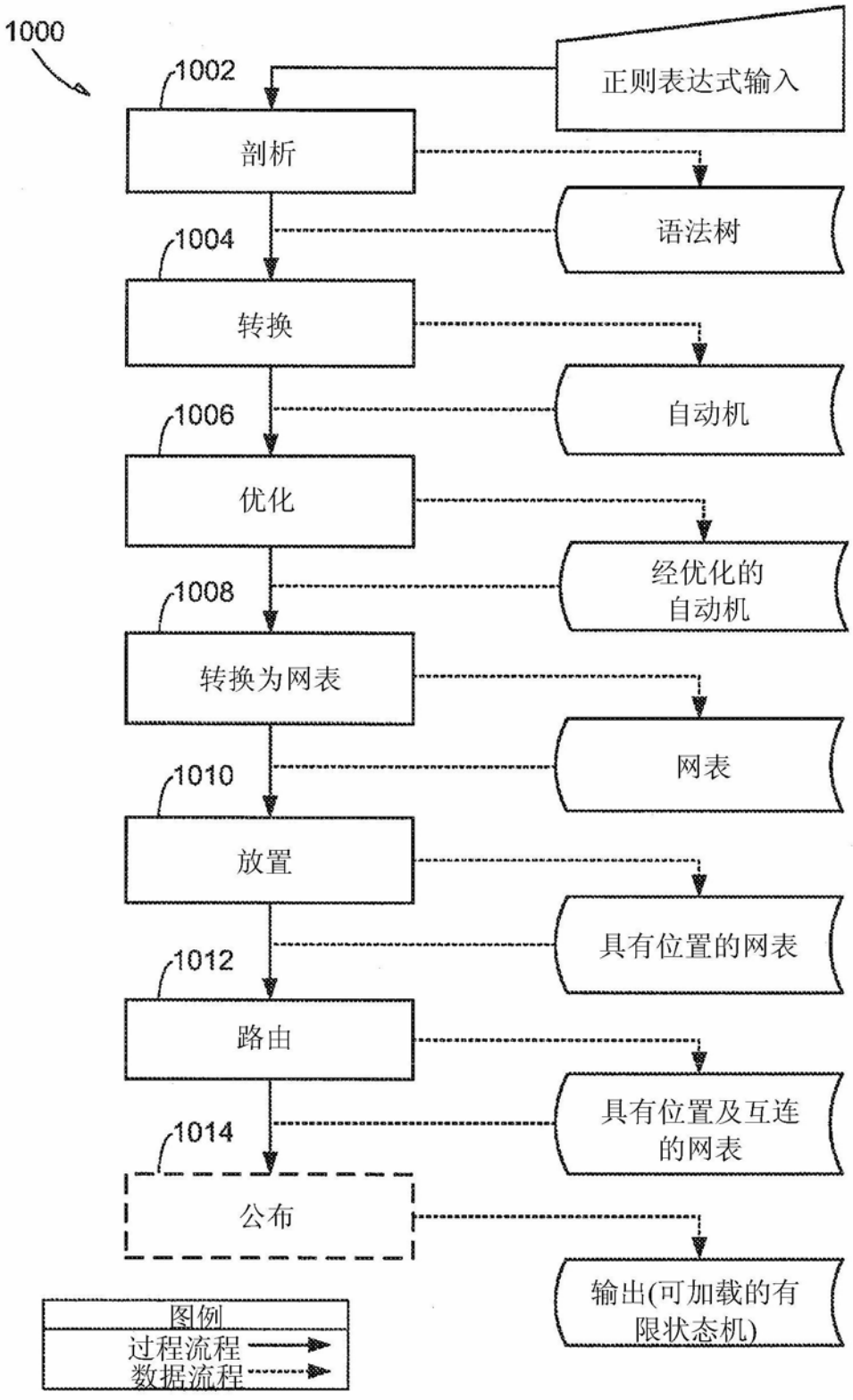

1000
1002
正则表达式输入
剖析
语法树
1004
转换
自动机
1006
优化
经优化的
自动机
1008
转换为网表
网表
1010
放置
具有位置的网表
1012
路由
具有位置及互连
的网表
1014
公布
输出(可加载的有
限状态机)
图例
过程流程
数据流程

图12

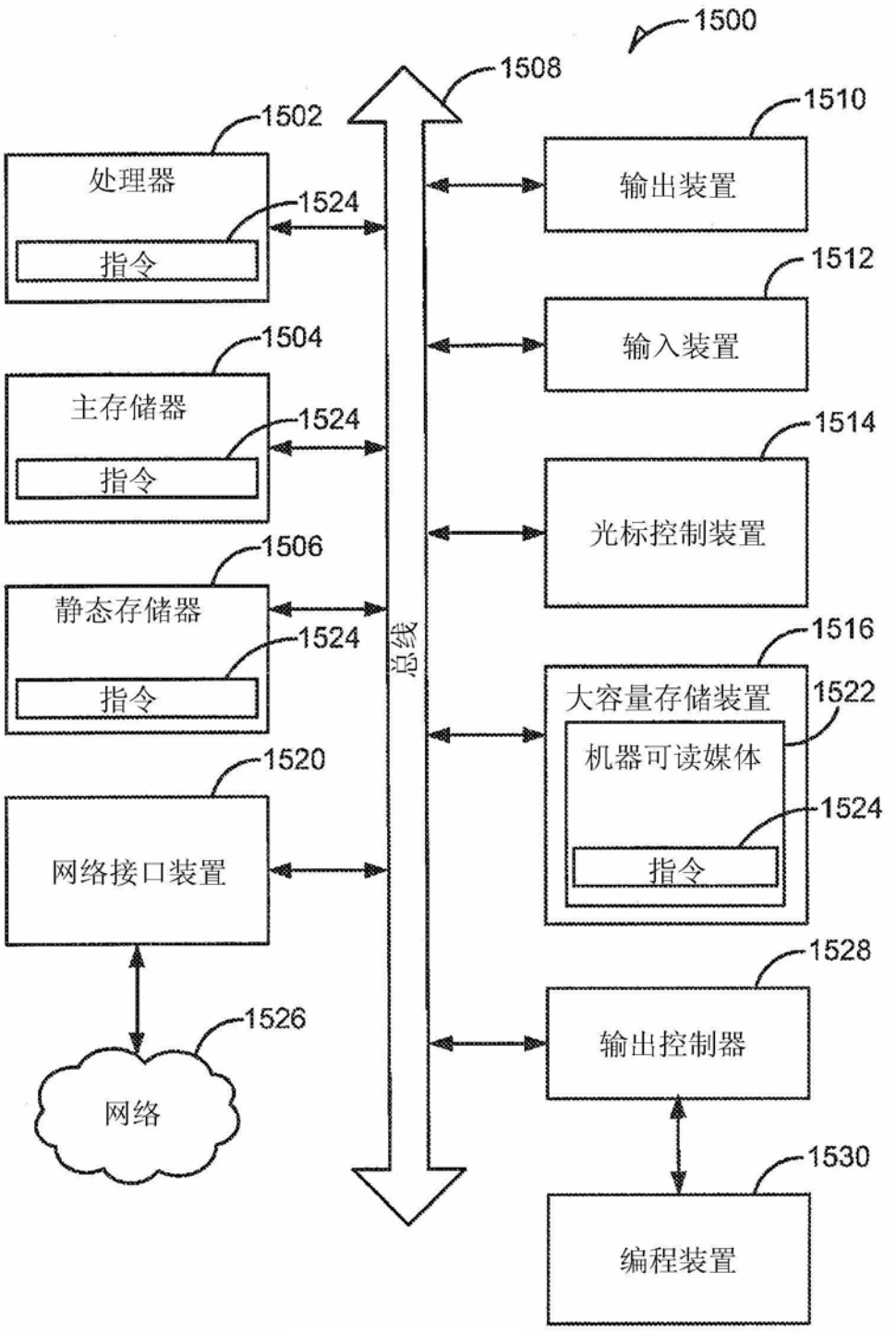

1500
1508
1502
处理器
1524
指令
1504
主存储器
1524
指令
1506
静态存储器
1524
指令
1520
网络接口装置
1526
网络
总线
1510
输出装置
1512
输入装置
1514
光标控制装置
1516
大容量存储装置
1522
机器可读媒体
1524
指令
1528
输出控制器
1530
编程装置

图13