



# [12] 发明专利申请公布说明书

[21] 申请号 200710162354.6

[43] 公开日 2008年6月11日

[11] 公开号 CN 101196947A

[22] 申请日 2007.9.29

[21] 申请号 200710162354.6

[30] 优先权

[32] 2006.12.4 [33] JP [31] 2006-327388

[71] 申请人 富士通株式会社

地址 日本神奈川县川崎市

[72] 发明人 加藤嘉之 青山久志

[74] 专利代理机构 北京三友知识产权代理有限公司  
代理人 黄纶伟 迟军

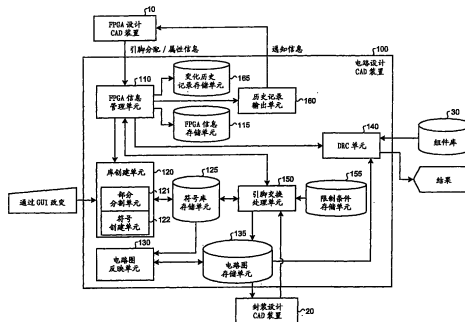
权利要求书4页 说明书21页 附图22页

## [54] 发明名称

电路设计支持装置和方法、以及印刷电路板的制作方法

## [57] 摘要

本发明涉及电路设计支持装置、电路设计支持方法、计算机产品以及印刷电路板的制作方法。一种电路设计 CAD 装置中含有的 FPGA 信息管理单元，其检索由 FPGA 设计 CAD 装置生成的 FPGA 信息，如引脚的分配信息和属性信息。库创建单元使用所述 FPGA 信息创建符号库。在创建符号库时，如果创建符号库的 FPGA 设置在电路图中，则库创建单元尽可能的管理，而不是改变电路图中现有符号库的部分划分和引脚布局。在设置符号库新创建的 FPGA 符号时，电路图反映单元在不改变现有布局的情况下设置所述符号。



1、一种支持设计电路的电路设计支持装置，在所述电路中将 PLD（可编程逻辑器件）用作组件，所述电路设计支持装置包括：

PLD 信息接收单元，其接收 PLD 信息，该 PLD 信息是通过使用 PLD 设计 CAD（计算机辅助设计）针对该 PLD 创建的设计信息；和

库创建单元，其通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

2、根据权利要求 1 所述的电路设计支持装置，其中所述库创建单元确定在电路图中是否布置有待创建符号库的所述 PLD 的符号，如果确定在所述电路图中布置有所述符号，则基于有关与在所述电路图中布置的所述符号相对应的符号库的信息，来在几乎不改变所述符号的情况下创建一符号库。

3、根据权利要求 2 所述的电路设计支持装置，其中如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则所述库创建单元基于有关与在所述电路图中布置的所述符号相对应的符号库的信息来执行部分分配。

4、根据权利要求 3 所述的电路设计支持装置，其中如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则所述库创建单元将布置在所述电路图中的所述符号所包含的逻辑引脚分配给与布置在所述电路图中的所述符号所在的部分相同的部分。

5、根据权利要求 4 所述的电路设计支持装置，其中如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则所述库创建单元将布置在所述电路图中的所述符号所包含的所述逻辑引脚布置在与布置在所述电路图中的所述符号所在的位置相同的位置。

6、根据权利要求 3 所述的电路设计支持装置，其中如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则所述库创建单元基于物理引脚名将布置在所述电路图中的所述符号所不包含的逻辑引脚分配给一部分。

7、根据权利要求6所述的电路设计支持装置，其中如果在所述电路图中布置有待创建符号库的所述PLD的所述符号，则所述库创建单元将布置在所述电路图中的所述符号所不包含的所述逻辑引脚布置在所述电路图中布置的所述符号未占用的位置。

8、根据权利要求1所述的电路设计支持装置，该电路设计支持装置还包括变化接收单元，其接收与所述符号库中的变化有关的信息。

9、根据权利要求8所述的电路设计支持装置，其中与变化有关的所述信息是与部分分配和引脚位置有关的信息。

10、根据权利要求1所述的电路设计支持装置，该电路设计支持装置还包括符号替换单元，其执行以下操作：

确定在电路图中是否布置有待创建符号库的所述PLD的符号，并且

如果确定在所述电路图中布置有所述符号，则用所述库创建单元针对其创建了所述符号库的符号替换布置在所述电路图中的所述符号。

11、根据权利要求10所述的电路设计支持装置，其中在所述库创建单元针对其创建了所述符号库的所述符号中所包含的多个引脚之外，如果具有与该引脚被替换之前所使用的逻辑引脚名不同的逻辑引脚名的引脚连接有一线路，则所述符号替换单元切断所述线路。

12、一种用于支持设计电路的方法，在该电路中将PLD（可编程逻辑器件）用作组件，所述方法包括以下步骤：

接收步骤，其接收PLD信息，该PLD信息是通过使用PLD设计CAD（计算机辅助设计）针对该PLD创建的设计信息；以及

创建步骤，其通过使用所述PLD信息来创建要在电路设计中使用的PLD的符号库。

13、根据权利要求12所述的方法，其中所述创建步骤包括以下步骤：

确定步骤，确定在电路图中是否布置有待创建符号库的所述PLD的符号；如果在所述确定步骤中确定在所述电路图中布置有所述符号，则基于有关与在所述电路图中布置的所述符号相对应的符号库的信息，来在几乎不改变所述符号的情况下创建一符号库。

14、根据权利要求13所述的方法，其中所述创建步骤包括以下步骤：

如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则基于有关与在所述电路图中布置的所述符号相对应的符号库的信息来执行部分分配。

15、根据权利要求 14 所述的方法，所述方法还包括替换步骤，所述替换步骤包括以下步骤：

确定在电路图中是否布置有待创建符号库的所述 PLD 的符号，并且

如果在所述确定步骤中确定在所述电路图中布置有所述符号，则用在所述创建步骤中创建了其符号库的符号替换布置在所述电路图中的所述符号。

16、一种计算机可读记录介质，其中存储有用于使计算机实现用于支持设计电路的方法的计算机程序，在所述电路中将 PLD（可编程逻辑器件）用作组件，所述计算机程序使所述计算机执行以下操作：

接收操作，其接收 PLD 信息，该 PLD 信息是通过使用 PLD 设计 CAD（计算机辅助设计）针对该 PLD 创建的设计信息；以及

创建操作，通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

17、根据权利要求 16 所述的计算机可读记录介质，其中所述创建操作包括：确定操作，确定在电路图中是否布置有待创建符号库的所述 PLD 的符号；如果在所述确定操作中确定在所述电路图中布置有所述符号，则基于有关与在所述电路图中布置的所述符号相对应的符号库的信息，来在几乎不改变所述符号的情况下创建一符号库。

18、根据权利要求 17 所述的计算机可读记录介质，其中所述创建操作包括以下操作：

如果在所述电路图中布置有待创建符号库的所述 PLD 的所述符号，则基于有关与在所述电路图中布置的所述符号相对应的符号库的信息来执行部分分配。

19、根据权利要求 16 所述的计算机可读记录介质，所述计算机程序还使所述计算机执行一替换操作，所述替换操作包括：

确定在电路图中是否布置有待创建符号库的所述 PLD 的符号，并且

如果在所述确定操作中确定在所述电路图中布置有所述符号，则在所述创建操作中创建了其符号库的符号替换布置在所述电路图中的所述符号。

20、一种制造印刷电路板的方法，所述方法被支持设计电路的电路设计支持装置使用，在所述电路中将 PLD（可编程逻辑器件）用作组件，所述方法包括以下步骤：

其接收 PLD 信息，该 PLD 信息是通过使用 PLD 设计 CAD（计算机辅助设计）针对该 PLD 创建的设计信息；以及

通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

## 电路设计支持装置和方法、 以及印刷电路板的制作方法

### 技术领域

本发明涉及用于支持（辅助）设计电路的技术，在所述电路中使用可编程逻辑器件（PLD）作为组件。

### 背景技术

在电路设计 CAD（计算机辅助设计）中，当将诸如 FPGA（现场可编程门阵列）的 PLD 用作组件时，电路设计者需要在设计 PLD 之后创建该 PLD 的符号，并将该符号登记在符号库中。但是，电路设计者的主要工作是通过协调组件来设计电路，因此大多数的电路设计者不习惯于创建组件的符号。因此，每次 PLD 设计改变时创建符号，都对电路设计者增加了沉重的负担。

因此，已经开发出了一种用于支持 PLD 的符号创建的技术。例如，日本专利申请特开 2006-79447 号公开了一种 FPGA 设计支持装置，该装置基于有关 FPGA 的引脚布局的信息而自动创建 FPGA 库。

但是，存在如下问题：尽管由 FPGA 设计支持装置创建了 FPGA 库，但是在设计电路的过程中，每次改变 FPGA 时都需要替换电路图中的 FPGA 符号。因而，在 FPGA 设计支持装置创建的 FPGA 符号中，在大多数情况下由于 FPGA 的变化导致部分划分和引脚布局发生变化，从而可能需要彻底改变电路图。

### 发明内容

本发明的目的是至少部分地解决传统技术中存在的问题。

根据本发明的一个方面，提供了一种支持设计电路的电路设计支持装置，在所述电路中将 PLD 用作组件，该电路设计支持装置包括：PLD

信息接收单元,其接收 PLD 信息,该 PLD 信息是通过使用 PLD 设计 CAD 针对该 PLD 创建的设计信息;和库创建单元,其通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

根据本发明的另一方面,提供了一种用于支持设计电路的方法,在该电路中将 PLD 用作组件,该方法包括以下步骤:接收 PLD 信息,该 PLD 信息是通过使用 PLD 设计 CAD 针对该 PLD 创建的设计信息;以及,通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

还是根据本发明的又一方面,提供了一种制造印刷电路板的方法,所述方法被支持设计电路的电路设计支持装置使用,在所述电路中将 PLD 用作组件,所述方法包括以下步骤:接收 PLD 信息,该 PLD 信息是通过使用 PLD 设计 CAD 针对该 PLD 创建的设计信息;以及,通过使用所述 PLD 信息来创建要在电路设计中使用的 PLD 的符号库。

还是根据本发明的再一方面,提供了一种计算机可读记录介质,在其中存储有使计算机实现上述方法的计算机程序。

通过结合附图来阅读对本发明的当前的优选实施方式的以下详细描述,将会更好的理解本发明上述和其他的目的、特征、优点以及技术上和工业上的意义。

#### 附图说明

图 1 是用于说明根据本发明第一实施方式的 FPGA 协同设计概念的说明图;

图 2 是根据第一实施方式的 FPGA 协同设计系统的功能框图;

图 3 是用于说明电路设计的说明图;

图 4A 和图 4B 是用于说明 FPGA 的符号库的说明图;

图 5 是在 FPGA 信息存储单元中存储的 FPGA 信息的实施例的图;

图 6 是在符号库存储单元中存储的符号库的实施例的图;

图 7 是引脚交换的实施例的图;

图 8 是如何将引脚交换反映到电路图中的实施例的图;

图 9 是在限制条件存储单元中存储的限制条件的实施例的图;

图 10 是在变化历史记录存储单元中存储的变化历史记录的实施例的图；

图 11 是历史记录输出单元输出给 FPGA 设计 CAD 装置的通知信息的实施例的图；

图 12 是通知信息输出格式的图；

图 13 是电路设计 CAD 装置执行的用于创建符号库的处理和用于布置符号的处理的流程图；

图 14 是由 DRC 单元执行的用于检查输入/输出属性的处理的流程图；

图 15 是由 DRC 单元执行的用于检查差分信号的处理的流程图；

图 16 是由 DRC 单元执行的用于检查电源电压的处理的流程图；

图 17 是由引脚交换处理单元执行的用于引脚交换的处理的流程图；

图 18 是由历史记录输出单元执行的用于输出变化历史记录的处理的流程图；

图 19 是用于说明根据本发明第二实施方式的 FPGA 协同设计概念的说明图；

图 20 根据第二实施方式的 FPGA 协同设计系统的功能框图；

图 21 是网表检索单元检索的网表的实施例的图；

图 22 是网表转换单元输出的网表的实施例的图；

图 23 是临时库创建单元创建的临时库的实施例的图；

图 24 是由临时库创建装置执行的用于输出用于封装设计 CAD 的信息的处理的流程图；

图 25 是由临时库创建装置执行的用于反映封装的考虑结果的处理的流程图；和

图 26 是执行根据第一实施例的电路设计 CAD 程序的计算机的功能框图。

### 具体实施方式

下面参照附图详细描述本发明的示例性实施方式。顺便提及，在这



些实施方式中主要描述了将本发明应用于 FPGA 的情况。

首先，说明根据本发明的第一实施方式的 FPGA 协同设计的概念。图 1 是用于说明根据第一实施方式的 FPGA 协同设计的概念的说明图。如该图所示，在第一实施方式的 FPGA 协同设计中，以下三个部分相互协作以支持设计者：支持 FPGA 设计的 FPGA 设计 CAD 装置 10、支持印刷电路板的封装设计的封装设计 CAD 装置 20，以及作为电路设计支持装置的电路设计 CAD 装置 100。

具体地说，电路设计 CAD 装置 100 接收 FPGA 信息（例如 FPGA 设计 CAD 装置 10 生成的引脚布局），并创建符号库。在创建 FPGA 的符号库时，如果待创建符号库的 FPGA 被布置在电路图中，即，如果符号库是由于 FPGA 中的变化而新创建的，则电路设计 CAD 装置 100 通过尽可能地使用有关现有符号的信息（如部分分配和符号引脚的布局）来创建符号库。

如上所述，电路设计 CAD 装置 100 通过使用 FPGA 信息来创建 FPGA 的符号库，使得电路设计者不需要创建 FPGA 的符号库。从而，能够降低电路设计者的工作量。同样，当符号库是由于 FPGA 的变化而新创建的时，电路设计 CAD 装置 100 通过尽可能地使用有关现有符号的信息来创建符号库。因此，可以减少对电路图的修改，从而提高电路设计的效率。

此外，当执行 DRC（设计规则校验）时，电路设计 CAD 装置 100 通过参照 FPGA 信息（如 FPGA 设计 CAD 装置 10 生成的引脚输入/输出属性）来执行所述 DRC。例如，电路设计 CAD 装置 100 通过参照 FPGA 的引脚输入/输出属性，来针对各网络检查输出引脚的数量。按此方式，电路设计 CAD 装置 100 通过参照 FPGA 信息（如引脚输入/输出属性）来执行所述 DRC，从而可以更精确地执行所述 DRC。

此外，当在封装设计中发生引脚交换时，电路设计 CAD 装置 100 从封装设计 CAD 装置 20 中检索引脚交换信息，并将该引脚交换反映在符号库、电路图等中。另外，电路设计 CAD 装置 100 在限制条件（如引脚之间的线路的长度）下将所述引脚交换反映在封装设计中。按此方式，

电路设计 CAD 装置 100 也在该限制条件下将该引脚交换反映在封装设计中，从而可以消除电路设计信息和封装设计信息中的不一致性。

此外，电路设计 CAD 装置 100 记录封装设计中引脚交换的历史记录，并将该引脚交换的历史记录信息提供给 FPGA 设计 CAD 装置 10。按此方式，电路设计 CAD 装置 100 记录封装设计中的引脚交换的历史记录，并将该引脚交换的历史记录信息提供给 FPGA 设计 CAD 装置 10，从而可以确保 FPGA 设计、电路设计和封装设计之间的一致性。

下面，说明根据第一实施方式的 FPGA 协同设计系统的结构。图 2 是根据第一实施方式的 FPGA 协同设计系统的功能框图。如该图所示，FPGA 协同设计系统包括 FPGA 设计 CAD 装置 10、封装设计 CAD 装置 20 和电路设计 CAD 装置 100。

电路设计 CAD 装置 100 与 FPGA 设计 CAD 装置 10 和封装设计 CAD 装置 20 相协作地支持电路设计，在所述电路中将 FPGA 用作组件。图 3 是用于说明该电路设计的说明图。如该图所示，电路设计者将登记为与一组件相关联的一符号库的符号布置在电路图中，并将符号引脚相互连接，从而设计该电路。

但是，在 FPGA 的情况中，如果在将程序写入在符号上之前将该符号登记为符号库，则由于引脚可被用于输入或输出中的任一个，因此可将引脚定义为输入/输出引脚。因此，当使用登记的符号库时，用于输入的引脚位于右侧，反之，用于输出的引脚位于左侧，或者按照图 4A 所示不是依次地布置总线，结果电路图变得复杂。

因此，每次写入程序时都必须创建符号库。结果，在该情况下，每次写入程序时电路设计 CAD 装置 100 都创建 FPGA 的符号库。通过在每次写入程序时创建 FPGA 的符号库，如图 4B 所示，可以将用于输入的引脚布置在左侧，而总线也可以依次地设置。

回到对图 2 的说明，电路设计 CAD 装置 100 包括 FPGA 信息管理单元 110、FPGA 信息存储单元 115、库创建单元 120、符号库存储单元 125、电路图反映单元 130、电路图存储单元 135、DRC 单元 140、引脚交换处理单元 150、限制条件存储单元 155、历史记录输出单元 160 以及变化历

史记录存储单元 165。

FPGA 信息管理单元 110 是用于管理 FPGA 信息的处理单元。该 FPGA 信息管理单元 110 从 FPGA 设计 CAD 装置 10 输出的文件中检索 FPGA 信息，例如物理引脚和逻辑引脚之间的对应关系、引脚输入/输出属性、电压值、以及簇编号 (bank number)，并将检索到的 FPGA 信息存储在 FPGA 信息存储单元 115 中。

在该情况下，在 FPGA 信息管理单元 110 首先将 FPGA 信息存储在 FPGA 信息存储单元 115 中之后，当 FPGA 信息管理单元 110 检索到 FPGA 信息并重新在 FPGA 信息存储单元 115 中存储所检索的 FPGA 信息时，FPGA 信息管理单元 110 进一步将指示 FPGA 信息变化的变化历史记录存储在变化历史记录存储单元 165 中。

FPGA 信息存储单元 115 是在 FPGA 信息管理单元 110 的控制下，在其中存储 FPGA 信息的存储单元。图 5 是存储在 FPGA 信息存储单元 115 中的 FPGA 信息的实施例的图。如该图所示，FPGA 信息存储单元 115 在其中存储有物理引脚名、逻辑引脚名、输入/输出属性、簇编号、交换组数量、差分属性，以及相对于每个引脚的电源电压。

库创建单元 120 是利用存储在 FPGA 信息存储单元 115 中的 FPGA 信息来创建 FPGA 的符号库的处理单元，并将所创建的符号库存储在符号库存储单元 125 中。库创建单元 120 包括部分划分单元 121 和符号创建单元 122。部分划分单元 121 将电路图划分为多个部分。符号创建单元 122 创建被部分划分单元 121 划分的各个部分的符号。

部分划分单元 121 基于用户通过 GUI 指定的部分划分规则，将电路图划分为多个部分。在部分划分时，电路图被各簇编号、各簇组、各逻辑引脚名等划分为所述多个部分。另外，部分划分单元 121 基于输入/输出属性确定符号最右侧引脚和最左侧引脚的位置，并通过利用引脚属性进行排序来确定引脚的顺序。此外，部分划分单元 121 接收通过 GUI 来自用户的用于替换部分之间的引脚的规定，并替换该引脚。

另外，在创建符号库时，库创建单元 120 检查待创建符号的 FPGA 该符号是否被布置在电路图中。如果符号被布置在电路图中，则库创建

单元 120 通过参照有关所布置的符号的信息来创建符号库。

具体地说，库创建单元 120 通过参照作为现有符号库密钥的逻辑引脚名来执行部分分配。另外，库创建单元 120 创建符号库，在该符号库中，将符号引脚布置在与在现有符号库中在先引脚所在的位置相同的位置。即，库创建单元 120 将具有在现有符号中包含的逻辑引脚名的引脚分配到与现有部分中在先引脚所在的位置的相同部分的相同位置上，而将具有在现有符号中不包含的逻辑引脚名的引脚分配到与具有相同物理引脚名的在先引脚所在的部分相同的部分上。另外，库创建单元 120 将具有与在现有符号中使用的逻辑引脚名不同的逻辑引脚名的引脚布置在符号库上未占用的位置。如果在符号库上没有未占用的位置，则库创建单元 120 沿向下方向扩展符号库的大小，并将引脚布置在扩展的位置。

当待创建符号的 FPGA 的符号被布置了在电路图中时，库创建单元 120 通过参照有关所布置的符号的信息来创建符号库，从而可以减少由于 FPGA 设计的变化而导致的电路图的修改。

符号库存储单元 125 是在其中存储有 FPGA 的符号库的存储单元。图 6 是存储在符号库存储单元 125 中的符号库的实施例的图。如该图所示，符号库存储单元 125 在其中存储有：与库名、创建的日期/时间、版本、占用区域、图表数量以及符号引脚的数量有关的信息；与形成符号的各图形有关信息；以及与各引脚有关的信息。

电路图反映单元 130 是用于进行如下替换的处理单元，即，在电路图中布置有 FPGA 的符号（其中由库创建单元 120 创建了符号库）的情况下，用新生成的符号替换所布置的符号。如果线路连接到具有与在所述替换之前使用的在先逻辑引脚名不同的逻辑引脚名的引脚，则电路图反映单元 130 切断该线路。

如果线路连接到具有与在所述替换之前使用的在先逻辑引脚名不同的逻辑引脚名的引脚，则电路图反映单元 130 切断该线路，从而可以减少由于 FPGA 设计的变化引起的电路图修改的遗漏。

电路图存储单元 135 是在其中存储有有关其中布置有多个组件的电路图的信息的存储单元。如果在电路图中布置有 FPGA 的符号（其中库

创建单元 120 创建了符号库), 则电路图存储单元 135 通过电路图反映单元 130 更新。

DRC 单元 140 是执行 DRC 的处理单元。DRC 单元 140 通过参照由 FPGA 信息管理单元 110 管理的 FPGA 信息以及存储在部件库 30 中的信息, 来执行 DRC。具体地说, DRC 单元 140 检查输入/输出属性、差分信号、电源电压等等。DRC 单元 140 通过参照 FPGA 信息来执行 DRC, 从而可以精确地执行与 FPGA 相关的 DRC。

引脚交换处理单元 150 是接收封装设计 CAD 装置 20 输出的引脚交换信息, 并将在封装设计中执行的引脚交换反映到 FPGA 信息、符号库、以及电路图的处理单元。在 FPGA 中, 可通过在程序进行写入来改变组件内部的运行, 这样, 在封装设计阶段执行 FPGA 组件的引脚替换(引脚交换)以容易地分配引脚。因此, 引脚交换处理单元 150 执行在电路设计中将引脚交换反映在封装设计中的处理。

图 7 是引脚交换的实施例的图。如该图所示, 当连接在 FPGA 与其他组件之间的线路交叉时, 可以通过 FPGA 的引脚交换来消除线路的交叉。图 8 是如何将引脚交换反映在电路图实施例的图。如该图所示, 在该电路图中, 在物理引脚名分别是“D1”、“E1”、“F1”和“G1”的符号中改变引脚布局。

引脚交换处理单元 150 还在限制条件(如引脚之间的线路的长度)下将引脚交换反映在封装设计中。引脚交换处理单元 150 还在限制条件下将引脚交换反映在封装设计中, 从而可以确保设计信息在电路设计与封装设计之间的一致性。

此外, 引脚交换处理单元 150 指示 FPGA 信息管理单元 110 存储由于引脚交换而导致的 FPGA 信息的变化历史记录。从而, FPGA 信息管理单元 110 将变化历史记录存储在变化历史记录单元 165 中。

限制条件存储单元 155 是在其中存储有与电路设计有关的限制条件(例如引脚之间的线路长度)的存储单元。图 9 是存储在限制条件存储单元 155 中的限制条件的实施例的图。如该图所示, 限制条件存储单元 155 在其中存储有与引脚之间的线路长度有关的限制条件。例如, 作为限

制条件存储有：具有组件“IC1”的物理引脚名“G1”的引脚与具有组件“IC2”的物理引脚名“2”的引脚之间的线路的长度是 50 mm 或以下。

历史记录输出单元 160 是以能够输入到 FPGA 设计 CAD 装置 10 的形式将 FPGA 信息（该信息在由引脚交换处理单元 150 执行的反映引脚交换的处理中发生改变）的变化历史记录作为通知信息输出给文件的处理单元。

变化历史记录存储单元 165 是在其中存储 FPGA 信息的变化历史记录的存储单元，并通过 FPGA 信息管理单元 110 进行管理。图 10 是在变化历史记录存储单元 165 中存储的变化历史记录的实施例的图。如该图所示，变化历史记录存储单元 165 在其中存储有有关处理的日期/时间以及每次执行引脚交换时交换的引脚的变化信息。另外，变化历史记录存储单元 165 在其中存储有每次历史记录输出单元 160 输出变化历史记录的处理的日期/时间，以及每次 FPGA 信息管理单元 110 从 FPGA 设计 CAD 装置 10 中检索 FPGA 信息的处理的日期/时间。

图 11 是历史记录输出单元 160 输出给 FPGA 设计 CAD 装置 10 的通知信息的实施例的图。如该图所示，历史记录输出单元 160 针对各交换的引脚输出引脚交换后的物理引脚名和改变的逻辑引脚名作为通知信息。图 12 是通知信息的输出格式的实施例的图。

按此方式，变化历史记录存储单元 165 在其中存储 FPGA 信息的变化历史记录，并且历史记录输出单元 160 以能输入到 FPGA 设计 CAD 装置 10 的形式将该变化历史作为通知信息输出给文件。这样，可以确保设计信息在封装设计、电路设计、电路设计以及 FPGA 设计之间的一致性。

接下来说明电路设计 CAD 装置 100 所执行的创建符号库的处理过程以及布置符号的处理。图 13 是通过电路设计 CAD 装置 100 执行的创建符号库的处理过程和布置符号的处理的流程图。

如该图所示，在电路设计 CAD 装置 100 中，FPGA 信息管理单元 110 从 FPGA 设计 CAD 装置 10 输出的文件中检索 FPGA 信息（例如引脚分配信息和属性信息），并在 FPGA 信息存储单元 115 中存储检索到的 FPGA 信息（步骤 S101）。

接下来，库创建单元 120 确定在电路图中是否布置有与 FPGA 信息管理单元 110 检索的 FPGA 信息相对应的符号（步骤 S102）。如果符号未布置在电路图中，则库创建单元 120 通过使用用户指定的部分划分规则将电路图划分为多个部分（步骤 S103），并根据用于创建符号的预定规则（例如，基于输入/输出属性将引脚分配到右侧或左侧）来确定符号引脚的位置（步骤 S104）。

如果在电路图中布置有符号，则库创建单元 120 通过参照先前执行的部分分配，将具有与在先引脚的逻辑引脚名相同的逻辑引脚名的引脚分配给与该在先引脚所在的部分相同的部分（步骤 S105）。如果一引脚具有未被在先引脚使用的逻辑引脚名，则库创建单元 120 将该引脚分配给与具有相同物理引脚名的在先引脚所在的部分相同的部分（步骤 S106）。然后，将具有与在先引脚的逻辑引脚名相同的逻辑引脚名的引脚布置在与在该先引脚所处的位置相同的位置（步骤 S107），而将具有未被在先引脚使用的逻辑引脚名的引脚布置到所述符号上的未占用的位置（步骤 S108）。

库创建单元 120 接收通过 GUI 来自用户的用于改变部分分配或引脚位置的规定。如果规定进行改变，则库创建单元 120 通过改变部分分配或引脚位置来创建符号库（步骤 S109），并将创建的符号库存储在符号库存储单元 125 中（步骤 S110）。

然后，电路图反映单元 130 确定在电路图中是否布置有在其中由库创建单元 120 创建了的符号库的 FPGA 的在先符号（步骤 S111）。如果在电路图中布置有在先符号，则电路图反映单元 130 将布置的符号替换为新创建的符号（步骤 S112）。如果要布置的引脚连接有线路，并且该引脚的逻辑引脚名与以前使用的引脚的逻辑引脚名不同，则电路图反映单元 130 切断该线路（步骤 S113）。

接下来，通过用户指定的组件输入功能将所创建的符号布置在电路图中（步骤 S114）。

按此方式，如果在电路图中设置有与通过 FPGA 信息管理单元 110 检索的 FPGA 信息相对应的符号，则库创建单元 120 通过参照在先创建

的符号库来创建符号库，并且电路图反映单元 130 将所布置的符号替换为其中新创建了符号库的符号。这样，可以减少由于 FPGA 设计的变化导致的电路图的修改。

顺便指出，在这种情况下，如果在电路图中布置有待创建符号库的 FPGA，则库创建单元 120 参照在先创建的符号库。作为另外一种选择，如果在符号库存储单元 125 中存储有待创建符号库的 FPGA 的符号库，则库创建单元 120 能够参照在先创建的符号库。

接下来说明 DRC 单元 140 执行的用于检查输入/输出属性的处理。图 14 是 DRC 单元 140 执行的检查输入/输出属性的处理的处理过程的流程图。

如该图所示，DRC 单元 140 关注单连接组的多个网络中的任一个，并获取有关在所关注的网络中包括的所有引脚的信息(步骤 S201)。然后，DRC 单元 140 关注获取了其信息的多个引脚中的任一个(步骤 S202)，并确定所关注的引脚是否用于 FPGA 组件(步骤 S203)。

结果，如果所关注的引脚是用于 FPGA 组件，则通过参照存储在 FPGA 信息存储单元 115 中的 FPGA 信息来检查引脚的输入/输出属性(步骤 S204)。如果所关注的引脚不是用于 FPGA 组件，则通过参照组件库 20 来检查引脚的输入/输出属性(步骤 S205)。然后，确定是否已检查了所有引脚的输入/输出属性(步骤 S206)。如果存在任一个未被检查的引脚，则系统控制返回到步骤 S202，并且关注未被检查的引脚以检查其输入/输出属性。

如果所有引脚的输入/输出属性都经过了检查，则确定所关注的网络是否包含两个或更多个输出引脚(步骤 S207)。如果包含有两个或更多个输出引脚，则将表示在输出引脚之间连接有网络的错误通知给用户(步骤 S208)。并且，确定所关注的网络是否不包含任何输出引脚(步骤 S209)。如果不包含有任何输出引脚，则将表示在所关注的网络中不存在输出引脚的错误通知给用户(步骤 S210)。如果只有一个引脚是输出引脚，则通知用户所关注的网络处于正常状态(步骤 S211)。

然后，确定是否已针对所有网络检查了输出引脚数量(步骤 S212)。



如果存在输出引脚数量未被检查的任何网络，则系统控制返回到步骤 S201，并且关注未检查输出引脚数量的网络以检查输出引脚数量。如果已针对所有网络检查了输出引脚数量，则终止检查输入/输出属性的程序。

按此方式，对于 FPGA 组件来说，DRC 单元 140 通过参照 FPGA 信息来检查引脚的输入/输出属性，从而可以在包括 FPGA 的电路中精确地检查输入/输出属性。

接下来说明 DRC 单元 140 执行的用于检查差分信号的处理。图 15 是 DRC 单元 140 执行的用于检查差分信号的处理的处理过程的流程图。

如该图所示，DRC 单元 140 关注多个网络中的任一个，并获取有关在所关注的网络中包括的所有引脚的信息（步骤 S301）。将正引脚数量（表示差分属性为正的引脚数量）和负引脚数量（表示差分属性为负的引脚数量）的初值清零（步骤 S302）。然后，关注获取了其信息的多个引脚中的任一个（步骤 S303），并确定所关注的引脚是否用于 FPGA 组件（步骤 S304）。

结果，如果所关注的引脚是用于 FPGA 组件，则通过参照在 FPGA 信息存储单元 115 中存储的 FPGA 信息来检查引脚的差分属性（步骤 S305）。如果所关注的引脚不是用于 FPGA 组件，则通过参照组件库 30 来检查引脚的差分属性（步骤 S306）。然后，如果差分属性为正，则正引脚的数量加“1”，或者如果差分属性为负，则负引脚的数量加“1”（步骤 S307）。

确定是否所有引脚的差分属性都经过了检查（步骤 S308）。如果存在在其差分属性未被检查的任何引脚，则系统控制返回到步骤 S303，并关注其差分属性未被检查的引脚以检查其差分属性。

如果所有引脚的差分属性都经过了检查，则确定正引脚的数量是否是正数，并且负引脚数量是否是正数，也就是说，确定是否具有正属性的引脚和具有负属性的引脚都存在在所关注的网络中（步骤 S309）。如果正引脚数量是正数，并且负引脚数量也是正数，则将表示将具有正属性的引脚连接到具有负属性的引脚的错误通知给用户（步骤 S310）。如果正引脚数量或负引脚数量中的仅任一个是正数，则通知用户关注网络处于

正常状态（步骤 S311）。

然后，确定是否已针对所有网络检查了差分信号（步骤 S312）。如果存在差分信号未被检查的任何网络，则系统控制返回到步骤 S301，并且关注未检查差分信号的网络以检查其差分信号。如果已针对所有网络检查了差分信号，则终止检查差分信号的处理。

按这种方式，对于 FPGA 组件来说，DRC 单元 140 通过参照 FPGA 信息来检查引脚的差分属性，从而可以精确地检查包括 FPGA 的电路中的差分信号。

接下来说明 DRC 单元 140 所执行的用于检查电源电压的处理。图 16 是 DRC 单元 140 执行的检查电源电压的处理的流程图。

如该图所示，DRC 单元 140 关注多个组件中的任一个（步骤 S401），并且还关注在所关注的组件中包括的多个引脚中的任一个（步骤 S402）。然后，确定所关注的引脚是否是电源引脚（步骤 S403）。如果所关注的引脚不是电源引脚，则系统控制进入到步骤 S410。

如果所关注的引脚是电源引脚，则确定所关注的引脚是否用于 FPGA 组件（步骤 S404）。如果所关注的引脚是用于 FPGA 组件，则通过参照在 FPGA 信息存储单元 115 中存储的 FPGA 信息来检查该引脚的电源电压（步骤 S405）。如果所关注的引脚不是用于 FPGA 组件，则通过参照组件库 30 来检查该引脚的电源电压（步骤 S406）。然后，检查连接有所关注的引脚的网络的电压值（步骤 S407），并确定该电压值是否与该引脚的电源电压相同（步骤 S408）。如果所述电压值与所述引脚的电源电压不同，则通知用户该电源电压与该电压值不同（步骤 S409）。

然后，确定是否已检查了所有引脚（步骤 S410）。如果存在未被检查的任何引脚，则系统控制返回到步骤 S402，并关注未检查的引脚以检查电源引脚的电压值。

如果已检查了所有的引脚，则确定是否已针对所有组件检查了电源电压（步骤 S411）。如果存在电源电压未被检查的任何组件，则系统控制返回到步骤 S401，并关注未检查电源电压的组件以检查其电源电压。如果确定已针对所有组件检查了电源电压，则终止检查电源电压的处理。

按此方式，对于 FPGA 组件来说，DRC 单元 140 通过参照 FPGA 信息来检查电源引脚的电压值，从而可以在包括 FPGA 的电路中精确地检查电源电压。

接下来说明引脚交换处理单元 150 执行的引脚交换的处理。图 17 是引脚交换处理单元 150 执行的引脚交换的处理的流程图。

如该图所示，引脚交换处理单元 150 检索封装设计 CAD 装置 20 创建的引脚交换信息（步骤 S501），并替换在其中执行了引脚交换的 FPGA 的符号库的物理引脚名（步骤 S502）。

随后，对在其中执行了引脚交换的 FPGA 中的 FPGA 信息中包括的逻辑引脚名和与逻辑有关的属性进行替换（步骤 S503），并将电路图中的符号更新为在其中替换了逻辑引脚名和与逻辑有关的属性的符号（步骤 S504）。对于具有限制条件的引脚，是在每次引脚交换时替换限制条件（步骤 S505）。

这样，对于具有限制条件的引脚来说，引脚交换处理单元 150 在每次执行引脚交换时替换限制条件，从而能够将封装设计 CAD 中的引脚交换精确地反映在有关电路设计的信息中。

下面说明历史记录输出单元 160 执行的用于输出变化历史记录的处理。图 18 是历史记录输出单元 160 执行的输出变化历史记录的处理的流程图。如该图所示，在从存储在变化历史记录存储单元 165 中的变化历史记录中检索到最新的 FPGA 信息之后，历史记录输出单元 160 搜索输出要通知给 FPGA 设计 CAD 装置 10 的通知信息的最后处理（步骤 S601）。

这样，对从输出通知信息的最新程序直到当前期间待执行引脚交换的引脚进行标记（步骤 S602）。将标记引脚的最新属性作为要通知给 FPGA 设计 CAD 装置 10 的通知信息输出（步骤 S603）。

也就是说，在 FPGA 信息管理单元 110 从 FPGA 设计 CAD 装置 10 检索到 FPGA 信息，并更新存储在 FPGA 信息存储单元 115 中的 FPGA 信息之后，历史记录输出单元 160 输出待执行交换的引脚的最新属性（这些属性尚未被通知）作为通知信息。

按此方式，历史记录输出单元 160 通过使用存储在变化历史记录存储单元 165 中的变化历史记录，来将待执行交换的引脚的最新属性作为通知信息输出给 FPGA 设计 CAD 装置 10，这样可以将封装设计中的引脚交换反映到 FPGA 设计信息中。

另外，在 FPGA 信息管理单元 110 从 FPGA 设计 CAD 装置 10 检索到 FPGA 信息，并更新存储在 FPGA 信息存储单元 115 中的 FPGA 信息之后，输出仅待执行引脚交换的引脚的最新属性（这些属性尚未被通知）作为通知信息。结果，可以避免输出废弃通知信息或避免通知信息的重叠，从而可以将封装设计中的引脚交换有效地反映到 FPGA 设计信息中。

如上所述，在第一实施方式中，包括在电路设计 CAD 装置 100 中的 FPGA 信息管理单元 110 检索 FPGA 信息，例如 FPGA 设计 CAD 装置 10 创建的引脚分配信息和属性信息，并且库创建单元 120 通过使用该 FPGA 信息来创建符号库。因此，电路设计者无需创建 FPGA 的符号库，从而可以降低电路设计者的工作量。

此外，在创建符号库时，如果在电路图中布置有待创建符号库的 FPGA，则库创建单元 120 设法尽可能地不改变布置在电路图中的现有符号库的部分分配和引脚布局。而且，当电路图反映单元 130 将新创建了其符号库的 FPGA 的符号布置在电路图中时，在不改变现有布局的情况下布置符号。这样，可以减少由于 FPGA 设计的变化而导致的电路图的修改。

另外，在第一实施方式中，当在电路设计 CAD 装置 100 中包括的 DRC 单元 140 执行 DRC 时，对于 FPGA，通过参照从 FPGA 设计 CAD 装置 10 中检索到的 FPGA 信息来检查引脚的属性等，并通过 FPGA 信息管理单元 110 将该属性存储在 FPGA 信息存储单元 115 中。从而，能够精确地执行 DRC。

此外，在第一实施方式中，在电路设计 CAD 装置 100 中包括的引脚交换处理单元 150 从封装设计 CAD 装置 20 中检索引脚交换信息，并除了符号库、FPGA 信息以及电路图以外，还将该引脚交换反映在限制条件中。从而，可以消除设计信息在电路设计与封装设计之间的不一致性。

另外，在第一实施方式中，在电路设计 CAD 装置 100 中包括的变化历史记录存储单元 165 在其中存储有 FPGA 信息的变化历史记录，历史记录输出单元 160 基于在变化历史记录存储单元 165 中存储的变化历史记录，将用于通知引脚交换的信息输出给 FPGA 设计 CAD 装置 10。这样，能够确保设计信息在封装设计、电路设计和 FPGA 设计之间的一致性。

在第一实施方式中，已考虑到如下的情况：基于通过使用 FPGA 组件进行电路设计的结果来进行印刷电路板的封装设计。预先考虑好 FPGA 设计者和封装设计者都期望的引脚分配，有助于大大缩短设计周期。因此，在本发明第二实施方式中，详细说明了支持 FPGA 设计者与封装设计者之间进行的协同设计的 FPGA 协同设计系统。

首先，说明根据第二实施方式的 FPGA 协同设计的概念。图 19 是用于说明根据第二实施方式的 FPGA 协同设计概念的说明图。如该图所示，在根据第二实施方式的 FPGA 协同设计中，作为协同设计支持装置的临时库创建装置 200 接收 FPGA 引脚信息（例如 FPGA 设计 CAD 装置 10 生成的引脚分配信息），并创建 FPGA 的临时库。在该情况下，临时库表示封装设计 CAD 装置 20 执行分配引脚时需要的组件形状类型库，并且是针对该 FPGA 临时创建的库。

临时库创建装置 200 从封装设计 CAD 装置 20 检索引脚交换信息，并将所检索的引脚交换信息反映在由其自身管理的 FPGA 信息中，并将该引脚交换信息通知给 FPGA 设计 CAD 装置 10。

按此方式，在第二实施方式中，临时库创建装置 200 接收 FPGA 设计 CAD 装置 10 生成的 FPGA 引脚信息，并创建针对该 FPGA 的临时组件形状类型库。这样，能够通过利用封装设计 CAD 装置 20 来考虑引脚分配。

接下来，说明根据第二实施方式的 FPGA 协同设计系统的结构。图 20 是根据第二实施方式的 FPGA 协同设计系统的结构的功能框图。如该图所示，FPGA 协同设计系统包括 FPGA 设计 CAD 装置 10、封装设计 CAD 装置 20、以及临时库创建装置 200。临时库创建装置 200 包括网表

检索单元 210、网表管理单元 220、网表转换单元 230，FPGA 设计 CAD 接口单元 240、FPGA 引脚信息管理单元 250、临时库创建单元 260、以及引脚交换处理单元 270。

网表检索单元 210 是检索用户创建的网表并将网表传送给网表管理单元 220 的处理单元。图 21 是网表检索单元 210 检索的网表的实施例的图。

如该图所示，网表包括定义组件的组件定义单元和定义网络的网络定义单元。在组件定义单元中，针对考虑用的组件描述组件名和组件库名。但是，对于 FPGA 组件来说，由于不存在组件库，所以以下用“FPGA/”描述其模块名（用于区别 FPGA 的名称）。

在网络定义单元中，针对各网络描述了网络名和连接到该网络的组件引脚。在这种情况下，以“(组件名).(组件引脚名)”的形式描述组件引脚。顺便提及的是，对于 FPGA 组件来说，将逻辑引脚名或物理引脚名描述为引脚名（用“%”标记物理引脚名）。

网表管理单元 220 是在其中存储并管理网表检索单元 210 检索的网表的管理单元。一旦接收到用户通过 GUI 输入的网表变化，网表管理单元 220 就改变网表。

网表转换单元 230 是将网表管理单元 220 管理的网表转换为能够输入封装设计 CAD 装置 20 的格式的处理单元。网表转换单元 230 在转换网表时参照由 FPGA 引脚信息管理单元 250 管理的 FPGA 信息。

图 22 是网表转换单元 230 输出的网表的实施例的图。如该图所示，网表包括针对各引脚的组件名、库名、组件端子号、引脚名、网名、交换组编号、以及差分类型。在这种情况下，组件端子号是分配给每个引脚的连续编号。

FPGA 设计 CAD 接口单元 240 是连接 FPGA 设计 CAD 装置 10 的接口部件。具体地说，FPGA 设计 CAD 接口单元 240 从 FPGA 设计 CAD 装置 10 中检索 FPGA 引脚信息，并将引脚交换信息提供给 FPGA 设计 CAD 装置 10。

FPGA 引脚信息管理单元 250 是在其中存储并由 FPGA 设计

CAD 接口单元 240 检索的 FPGA 引脚信息的管理单元。另外，一旦接收到通过 GUI 来自用户的改变引脚间隔或 FPGA 引脚信息的指令，FPGA 引脚信息管理单元 250 就改变 FPGA 信息。

临时库创建单元 260 是针对 FPGA 组件、利用 FPGA 引脚信息管理单元 250 管理的 FPGA 引脚信息来创建临时库（即创建临时组件形状类型库）的处理单元。

图 23 是临时库创建单元 260 创建的临时库的实施例的图。如该图所示，在临时库中针对每个引脚描述了连接区（land）形状类型库名、X 坐标、Y 坐标、角度以及引脚名。顺便提及的是，对于连接区形状类型库名来说，使用了从用户接收到指令时存储在 FPGA 引脚信息管理单元 250 中的信息。

另外，在临时库中还描述了表示组件尺寸的区域。使用有关该区域的信息来在封装设计过程中计算组件之间的距离。顺便提及的是，组件的尺寸是由临时库创建单元 260 基于引脚间隔而计算的。

临时库创建单元 260 基于 FPGA 引脚信息创建临时库，从而能够考虑在封装设计 CAD 中的引脚分配。

引脚交换处理单元 270 是从封装设计 CAD 装置 20 中检索引脚交换信息并指示 FPGA 引脚信息管理单元 250 改变 FPGA 引脚信息的处理单元。FPGA 引脚信息管理单元 250 改变 FPGA 引脚信息，并指示 FPGA 设计 CAD 接口单元 240 将该引脚交换信息通知给 FPGA 设计 CAD 装置 10。另外，引脚交换处理单元 270 指示网表管理单元 220 来基于该引脚交换信息而改变网表。

接下来说明临时库创建装置 200 执行的用于输出用于封装设计 CAD 的信息的处理。图 24 是由临时库创建装置 200 执行的用于输出用于封装设计 CAD 的信息的处理的处理过程的流程图。

如该图所示，在临时库创建装置 200 中，FPGA 设计 CAD 接口单元 240 检索由 FPGA 设计 CAD 装置 10 创建的引脚分配信息，并将该引脚分配信息传送给 FPGA 引脚信息管理单元 250，然后，FPGA 引脚信息管理单元 250 创建 FPGA 引脚信息（步骤 S701）。

此外，网表检索单元 210 检索网表（步骤 S702），并将该网表传送给网表管理单元 220。当从用户接收到用于改变网表等指令时，网表管理单元 220 改变由其自身管理的网表。当从用户接收到引脚间隔等的规定时（步骤 S703），FPGA 引脚信息管理单元 250 改变由其自身管理的 FPGA 引脚信息。

然后，临时库创建单元 260 从 FPGA 引脚信息中获取引脚的坐标，并创建临时组件形状类型库（步骤 S704），而网表转换单元 230 转换网表（步骤 S705）。接着，网表转换单元 230 将转换的网表输出给一文件，并且临时库创建单元 260 将所创建的临时库输出给该文件（步骤 S706）。

按此方式，临时库创建装置 200 创建临时库，从而能够通过使用封装设计 CAD 装置 20 来考虑引脚分配。此外，一旦从用户接收到引脚间隔等的规定，FPGA 引脚信息管理单元 250 就改变由其自身管理的 FPGA 引脚信息。从而，用户可以考虑采用多种引脚间隔的引脚分配。

下面说明由临时库创建装置 200 执行的反映封装的考虑结果的处理。图 25 是由临时库创建装置 200 执行的反映封装的考虑结果的处理的处理过程的流程图。

如该图所示，在临时库创建装置 200 中，引脚交换处理单元 270 在封装设计 CAD 中检索引脚交换信息（步骤 S801），并替换在网表中包含的待交换引脚的网络（步骤 S802）。

然后，引脚交换处理单元 270 替换在 FPGA 信息中包含的逻辑引脚名和逻辑属性（步骤 S803），并且 FPGA 设计 CAD 接口单元 240 将有关所替换的引脚的信息输出给一文件（步骤 S804）。

按此方式，引脚交换处理单元 270 检索封装设计 CAD 中的引脚交换信息，并将该引脚交换反映在网表中和 FPGA 引脚信息中。然后，FPGA 设计 CAD 接口单元 240 将有关该引脚交换的信息输出给该文件。这样，可以将封装设计中的引脚交换反映到 FPGA 设计信息中。

如上所述，在第二实施方式中，FPGA 设计 CAD 接口单元 240 检索 FPGA 设计 CAD 装置 10 创建的引脚分配信息，而 FPGA 引脚信息管理单元 250 将由 FPGA 设计 CAD 接口单元 240 检索到的引脚分配信息作为



FPGA 引脚信息进行管理,并且临时库创建单元 260 利用 FPGA 引脚信息管理单元 250 管理的 FPGA 引脚信息来创建临时组件形状类型库,并以能够被封装设计 CAD 装置 20 读取的形式将该临时组件形状类型库输出给该文件。由此,能够通过使用封装设计 CAD 装置 20 来考虑早期的引脚分配,还可以缩短设计印刷电路板的周期。

在第一和第二实施方式中分别描述了电路设计 CAD 装置和临时库创建装置。作为另外一种选择,也可以通过用软件实现电路设计 CAD 装置和临时库创建装置的结构,从而实现分别具有与电路设计 CAD 装置和临时库创建装置相同的功能的电路设计 CAD 程序和临时库创建程序。因此,下面描述执行电路设计 CAD 程序的计算机。顺便提及的是,也可以通过类似的计算机执行临时库创建程序。

图 26 是执行根据第一实施方式的电路设计 CAD 程序的计算机 300 的功能框图。如该图所示,计算机 300 包括 RAM 310、CPU 320、HDD 330、LAN 接口 340、输入/输出接口 350 以及 DVD 驱动器 360。

RAM 310 是在其中存储有计算机程序、执行计算机程序的中间结果等的存储器。CPU 320 是从 RAM 310 读取程序并执行该程序的中央处理单元。HDD 330 是在其中存储程序 and 数据的磁盘装置。LAN 接口 340 是用于将计算机 300 经由网络连接到其他计算机的接口。输入/输出接口 350 是用于将计算机 300 连接到输入装置(例如,鼠标或键盘,以及显示装置)的接口。DVD 驱动器 360 是读取/写入 DVD 的装置。

将待由计算机 300 执行的电路设计 CAD 程序 311 存储在 DVD 中,并通过 DVD 驱动器 360 从 DVD 读取出,然后安装在计算机 300 上。作为另外一种选择,电路设计 CAD 程序 311 存储在例如经由 LAN 接口 340 与计算机 300 连接的其他计算机系统的数据库中,从数据库中读取出该电路设计 CAD 程序 311 并将其安装在计算机 300 上。然后,将所安装的电路设计 CAD 程序 311 存储在 HDD 330 中,通过 RAM 310 读出,然后由 CPU 320 执行。

本实施方式中说明了将 FPGA 用作组件的情况。但是,本发明不局限于上述情况。本发明也可以适用于通常将 PLD 用作组件的情况。

根据本发明的一个方面，可以通过使用有关在电路图中布置的符号的信息来创建符号库。这样，能够减少由 PLD 的设计改变而导致的电路图修改。

根据本发明的另一方面，当在电路图中布置的 PLD 的设计发生改变时，能够减少符号的改变。因此，可以减少由于 PLD 设计的改变而导致的电路图修改，从而提高了设计电路的效率。

还是根据本发明的还一方面，当在电路图中布置的 PLD 的设计发生改变时，电路设计者不必更换电路图中的符号。从而，能够减少由于 PLD 的设计改变而导致的电路图修改，从而提高了设计电路的效率。

还是根据本发明的再另一方面，能够避免由于 PLD 的设计改变导致的电路图的修改的遗漏。从而可以提高设计质量。

尽管出于完整并清楚的公开的目的根据具体实施方式描述了本发明，但附后的权利要求并不因此而受到限制，而应解释具体体现了本领域的技术人员可以想到的完全落入本文所阐述的基本教示内的所有的变型和替换结构。

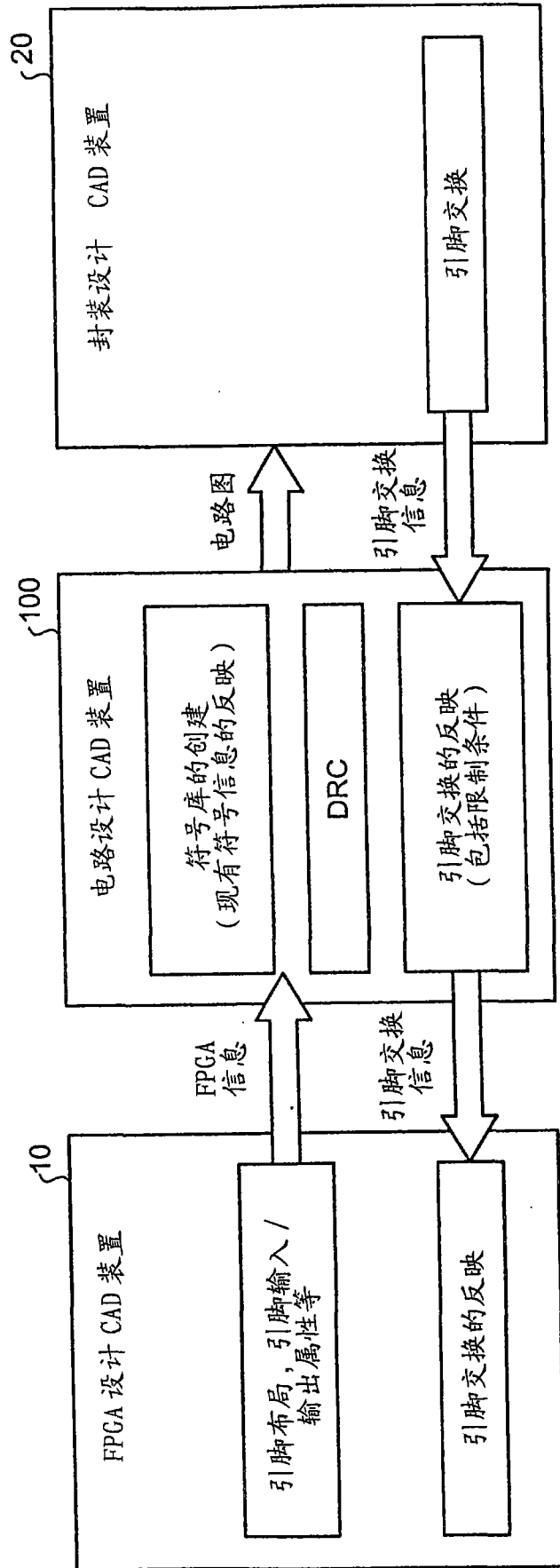


图 1

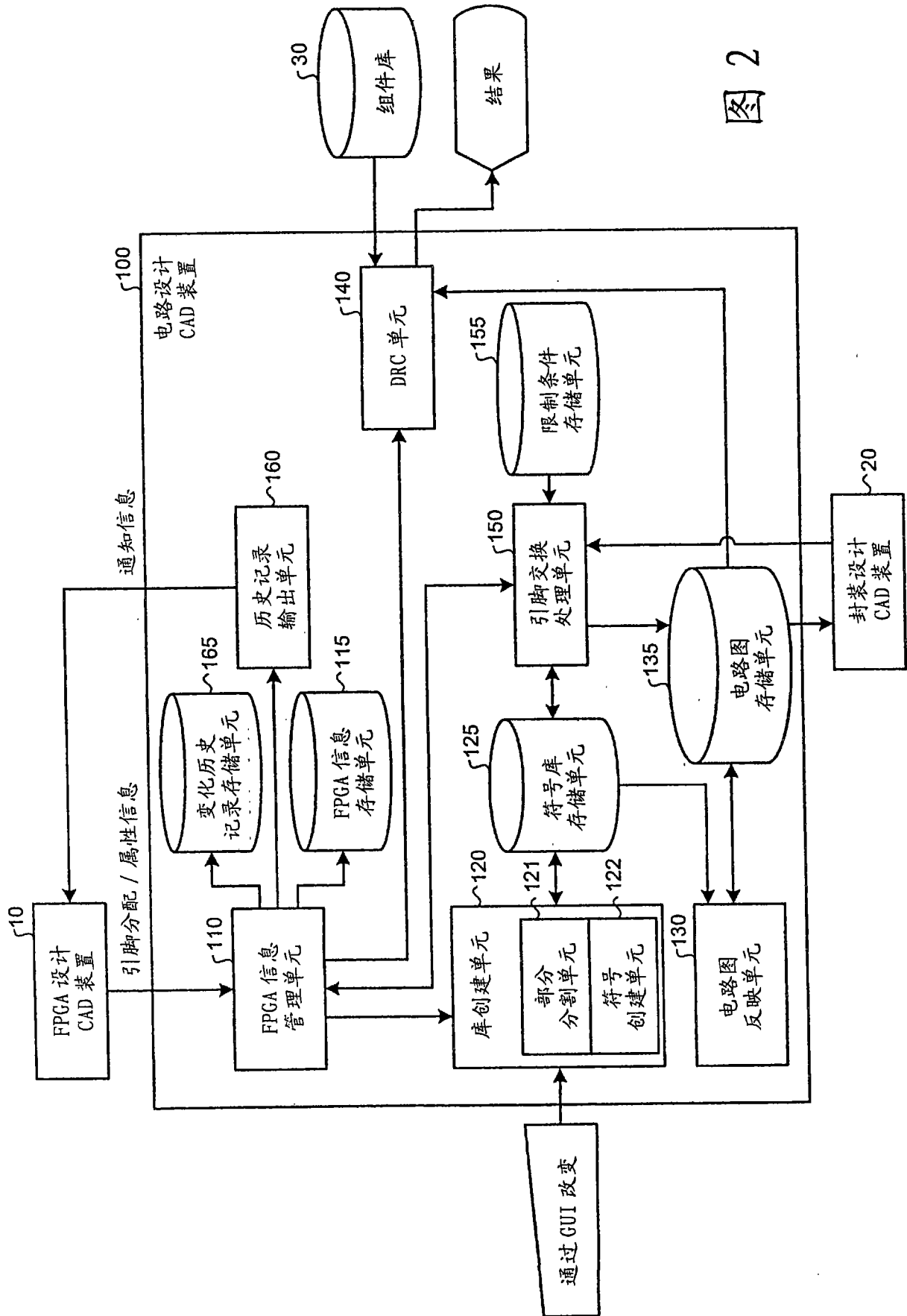


图 2

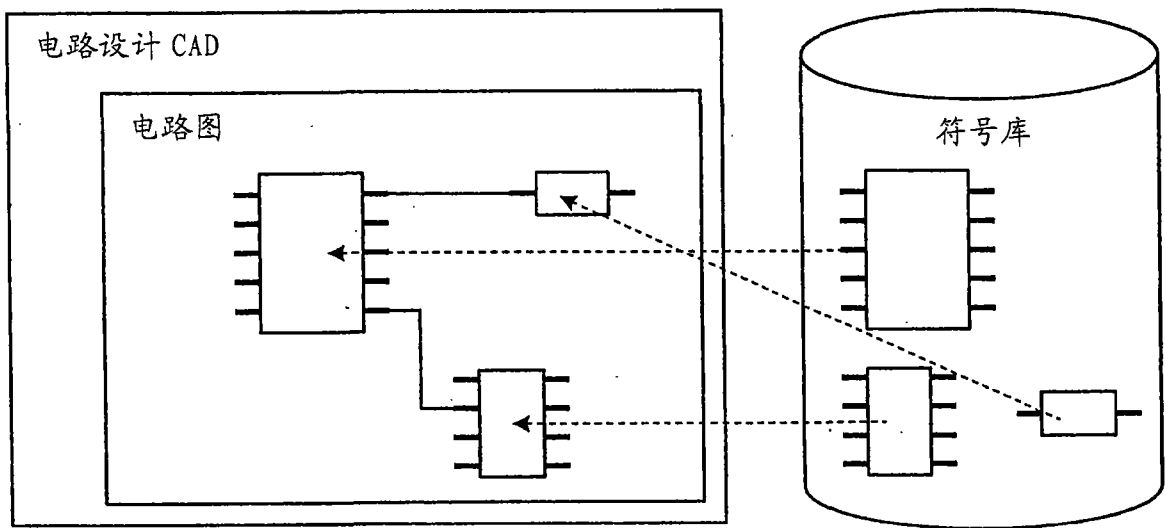


图 3

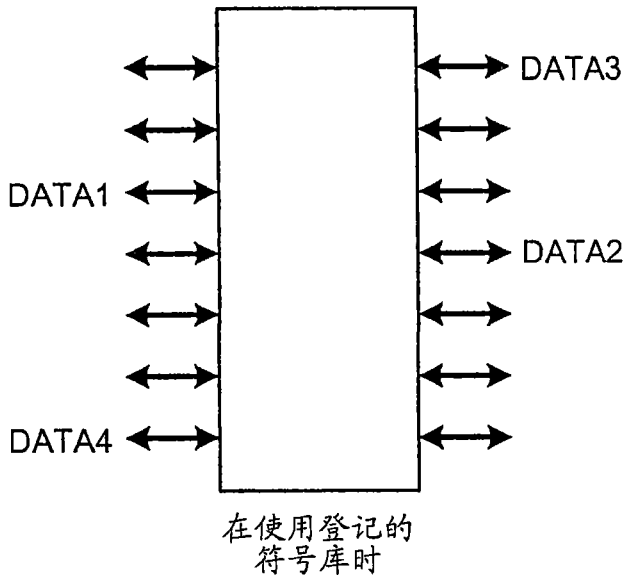


图 4A

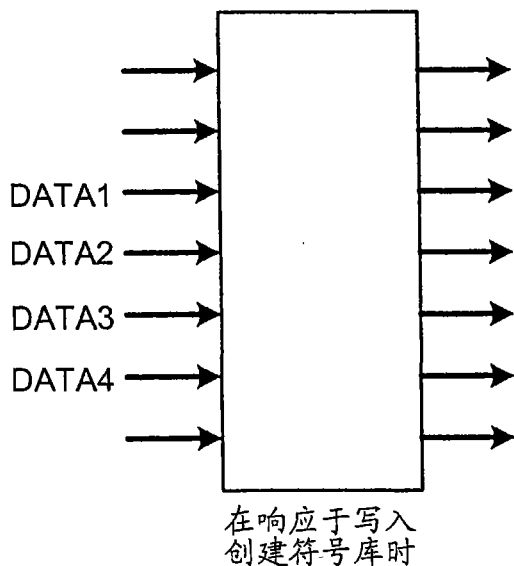


图 4B

物理引脚名	逻辑引脚名	输入 / 输出属性	簇编号	交换组编号	差分属性	电源电压
A0	ADRS1	输入	1	1		
A1	VCCIO	电源				3.3V
A2	ADRS2	输入	1	1		
A3	GND	接地				
A4	DATA1	输出	1	1	POSI	
A5	DATA2	输出	1	1	NEGA	
A6	VCCINT	电源				1.5V
C10	ADRSB1	输入	2	2		
C11	ADRSB2	输入	2	2		
C12	ADRSB3	输入	2	2		

图 5

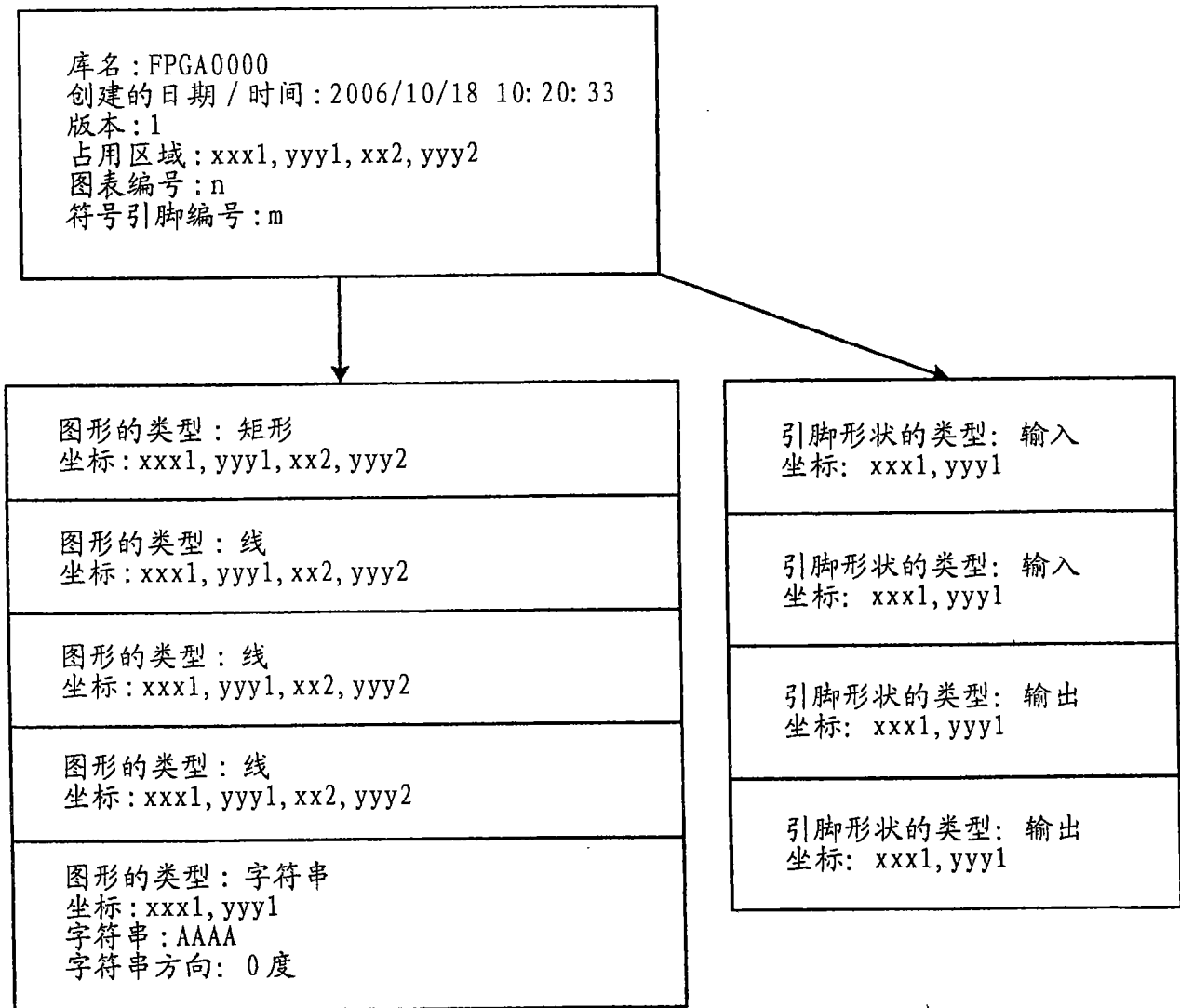


图 6



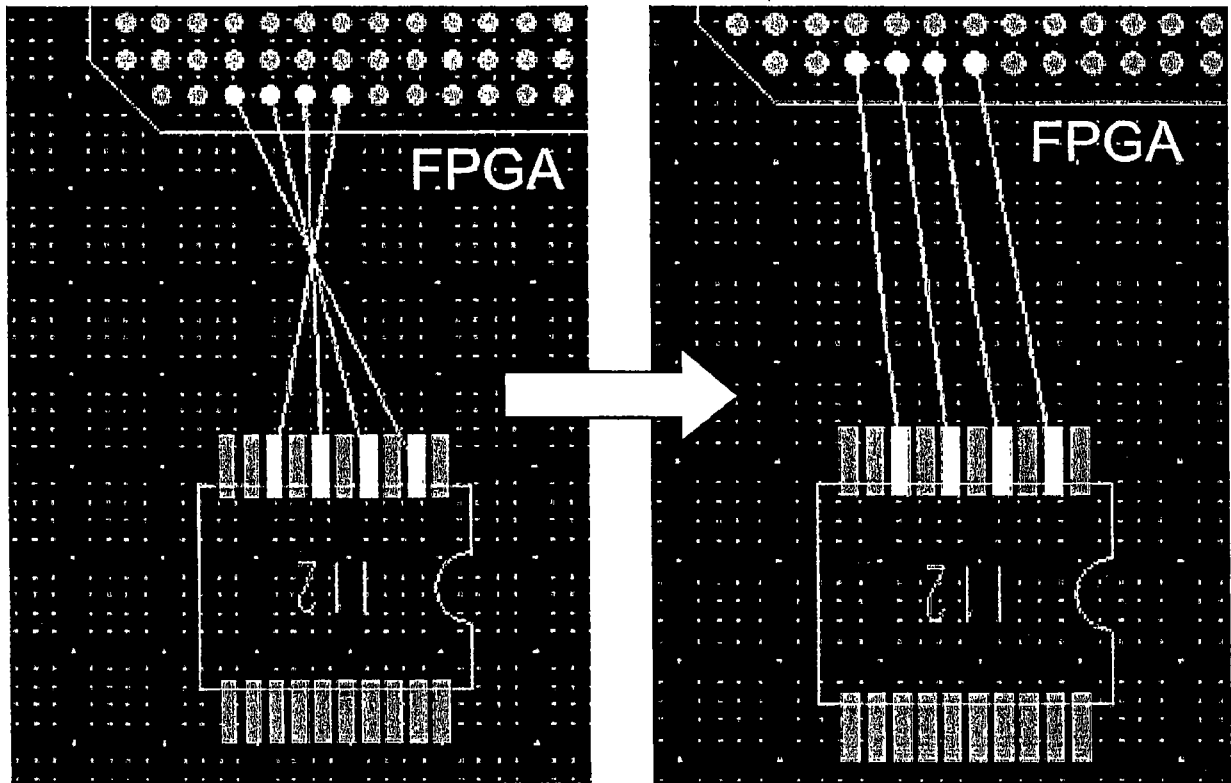


图 7



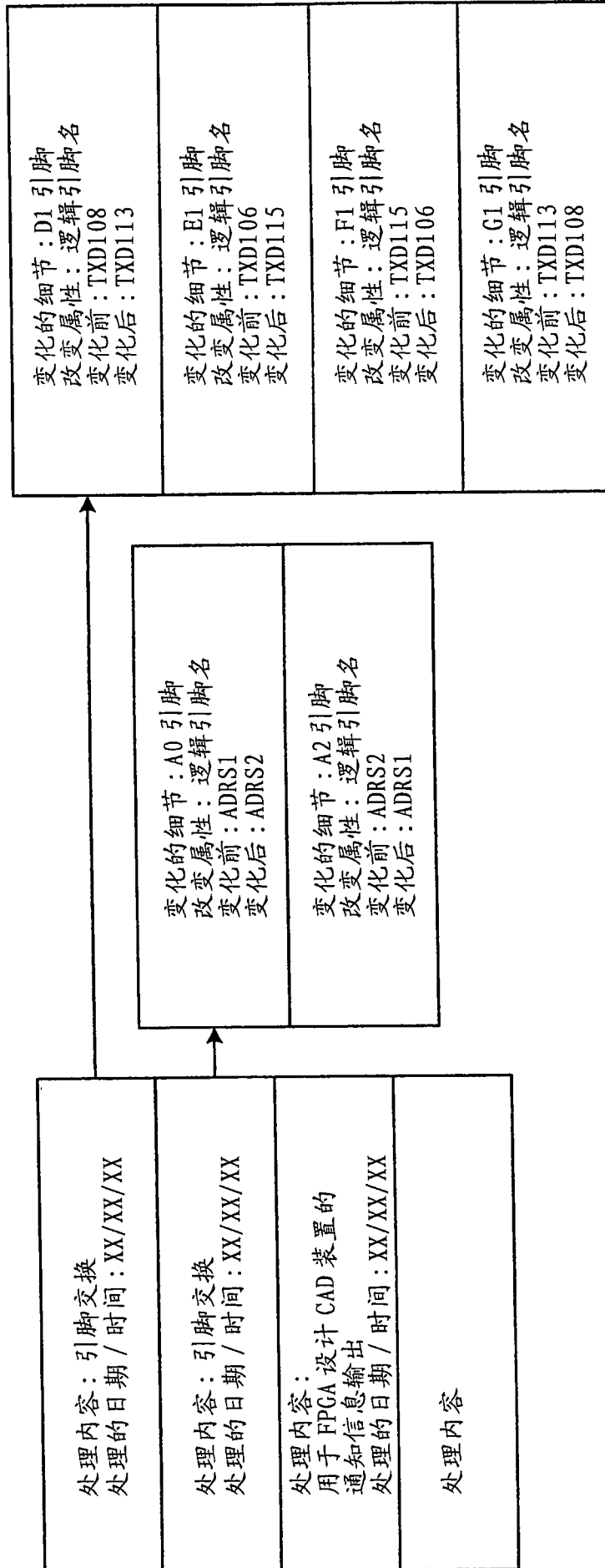


图 10

物理引脚名:A0 逻辑引脚名:ADRS2
物理引脚名:A2 逻辑引脚名:ADRS1
物理引脚名:D1 逻辑引脚名:TXD113
物理引脚名:E1 逻辑引脚名:TXD115
物理引脚名:F1 逻辑引脚名:TXD106
物理引脚名:G1 逻辑引脚名:TXD108

图 11

```
NET "TXD113" LOC="D1";  
NET "TXD115" LOC="E1";  
NET "TXD106" LOC="F1";  
NET "TXD108" LOC="G1";
```

图 12

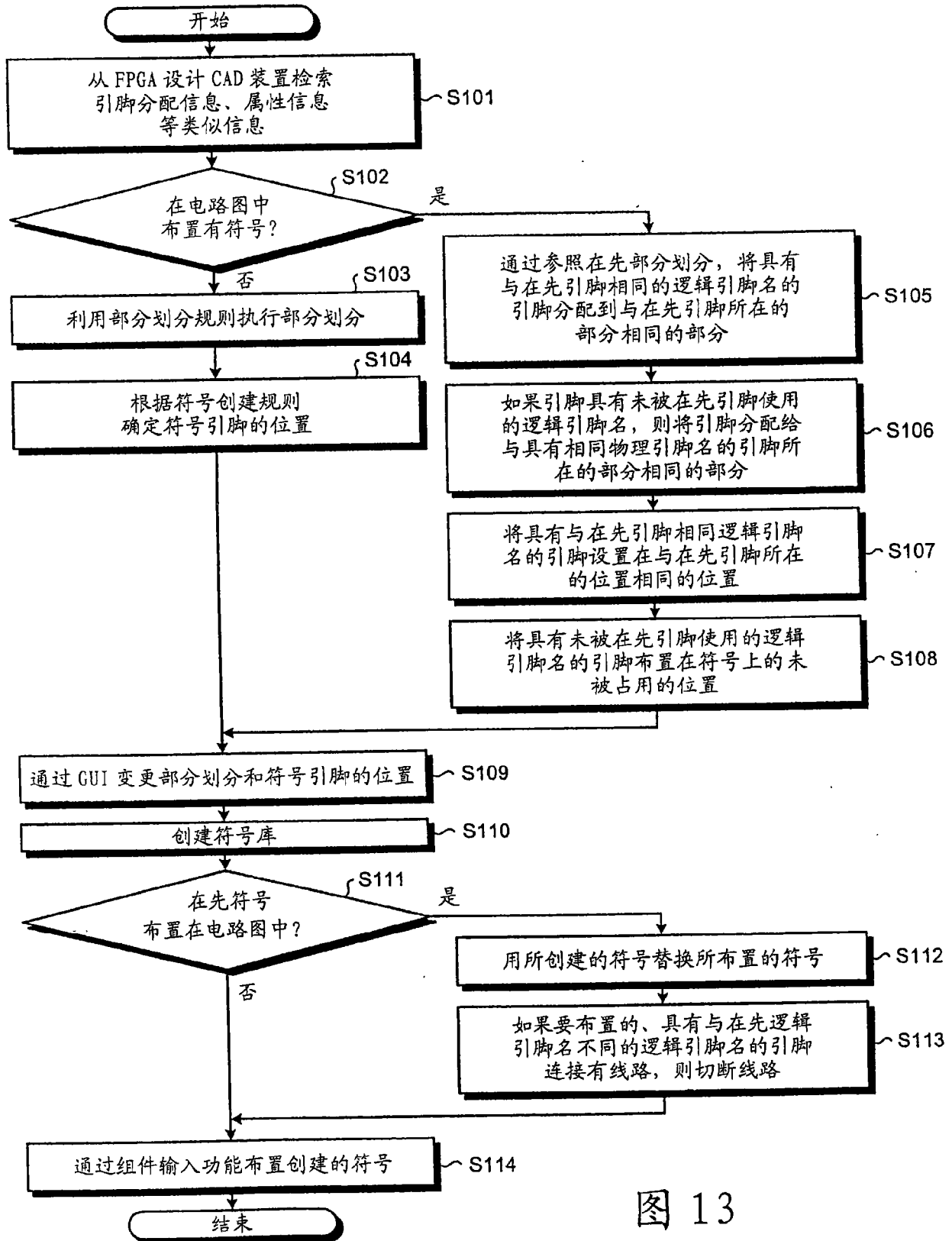


图 13

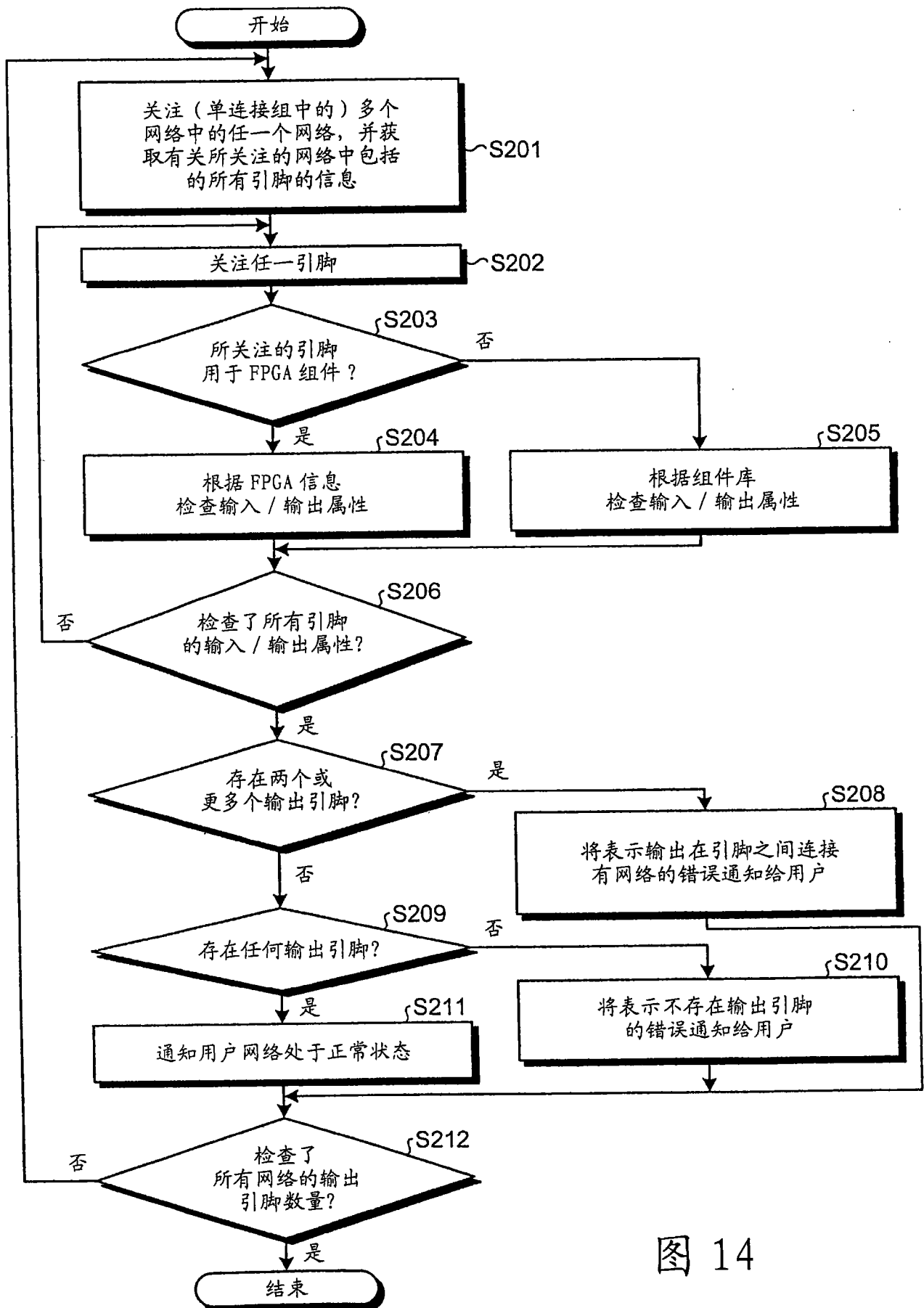


图 14

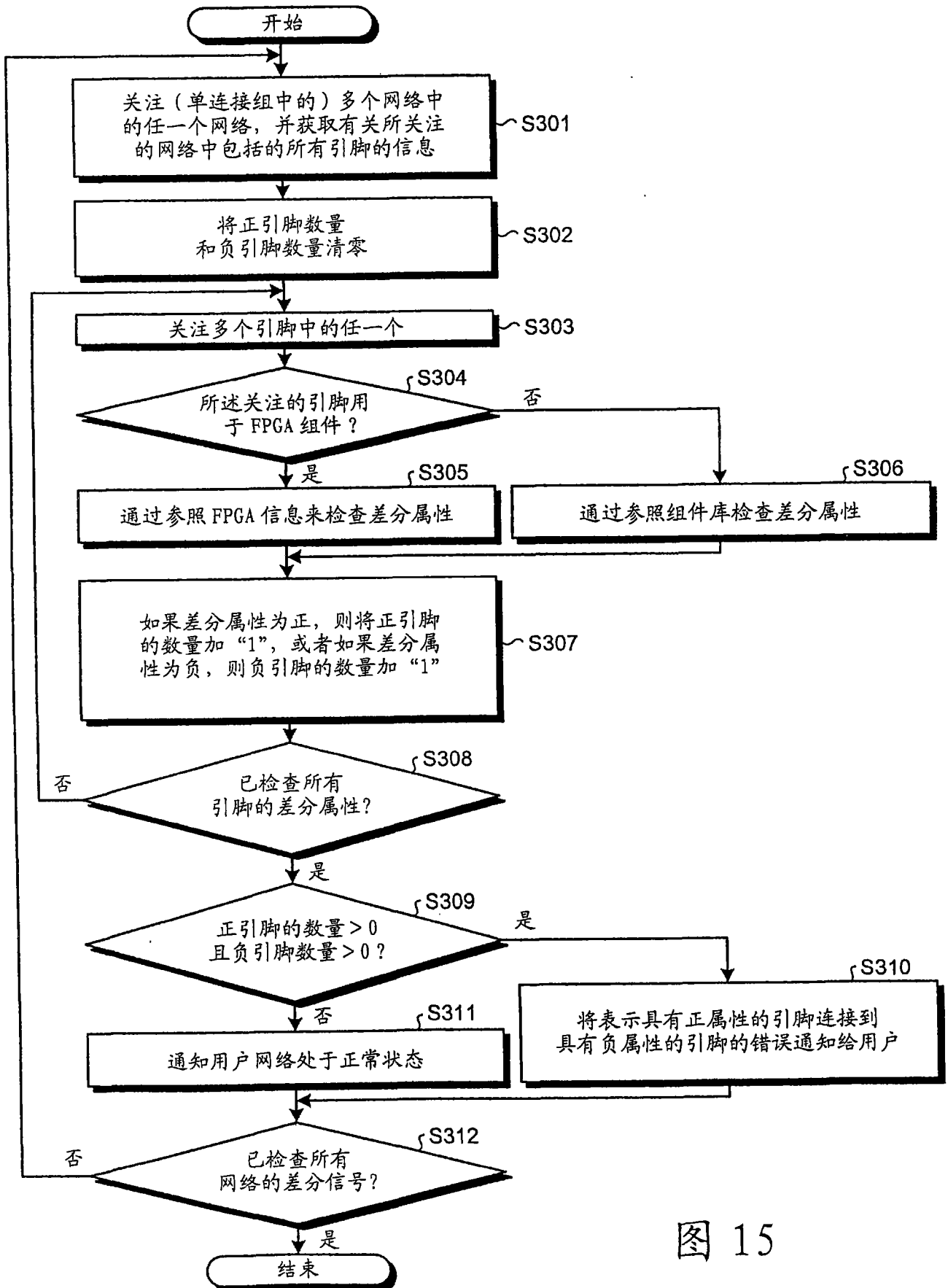


图 15

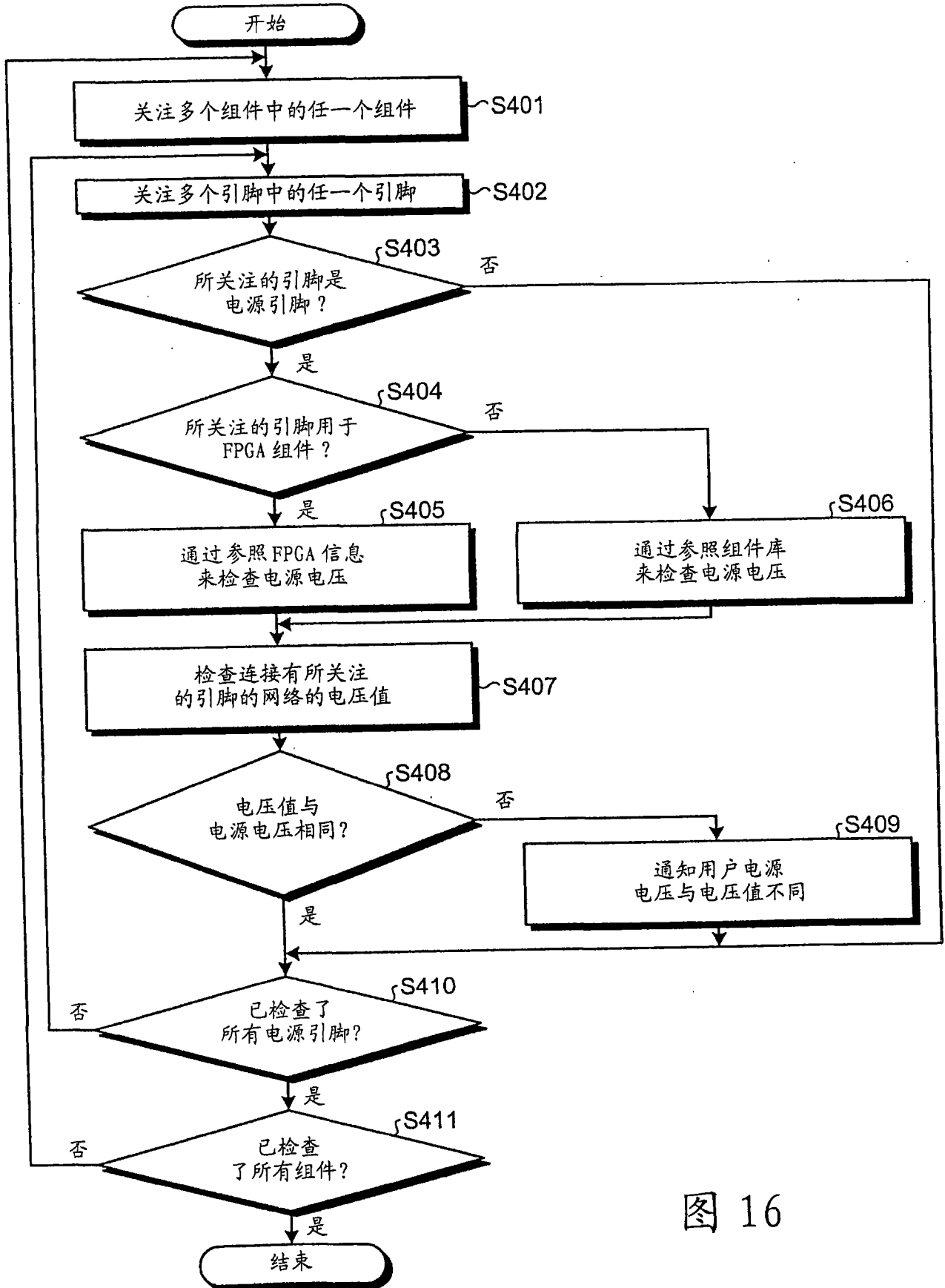


图 16



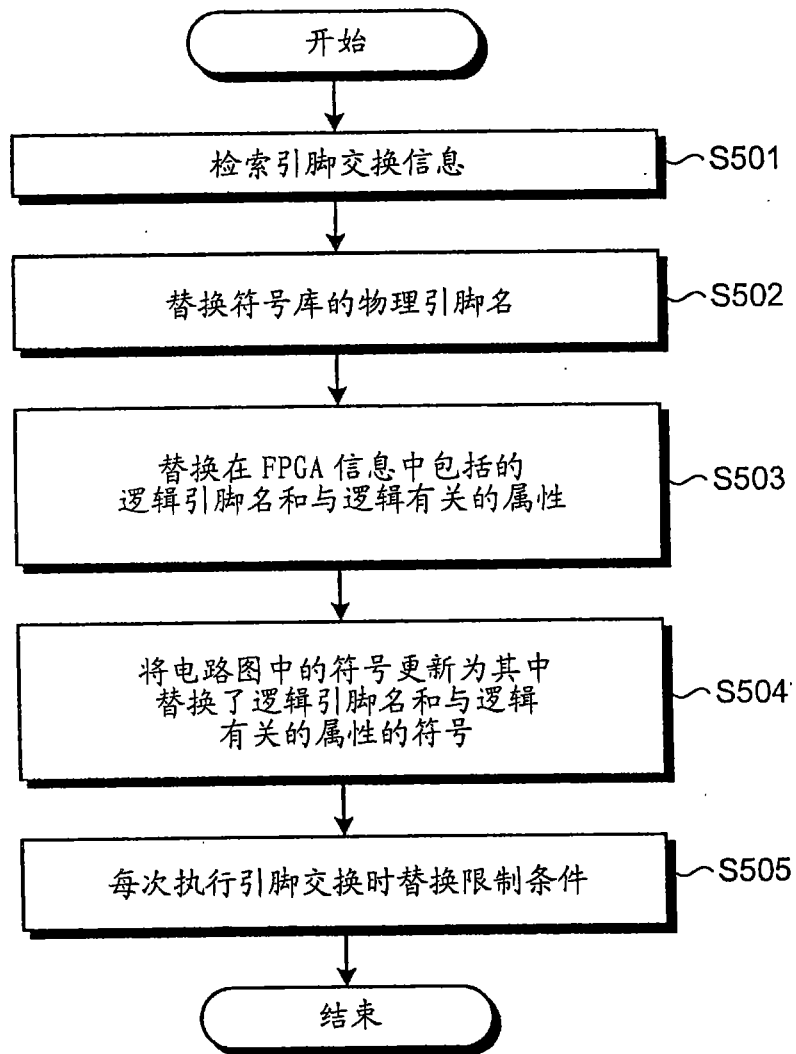


图 17

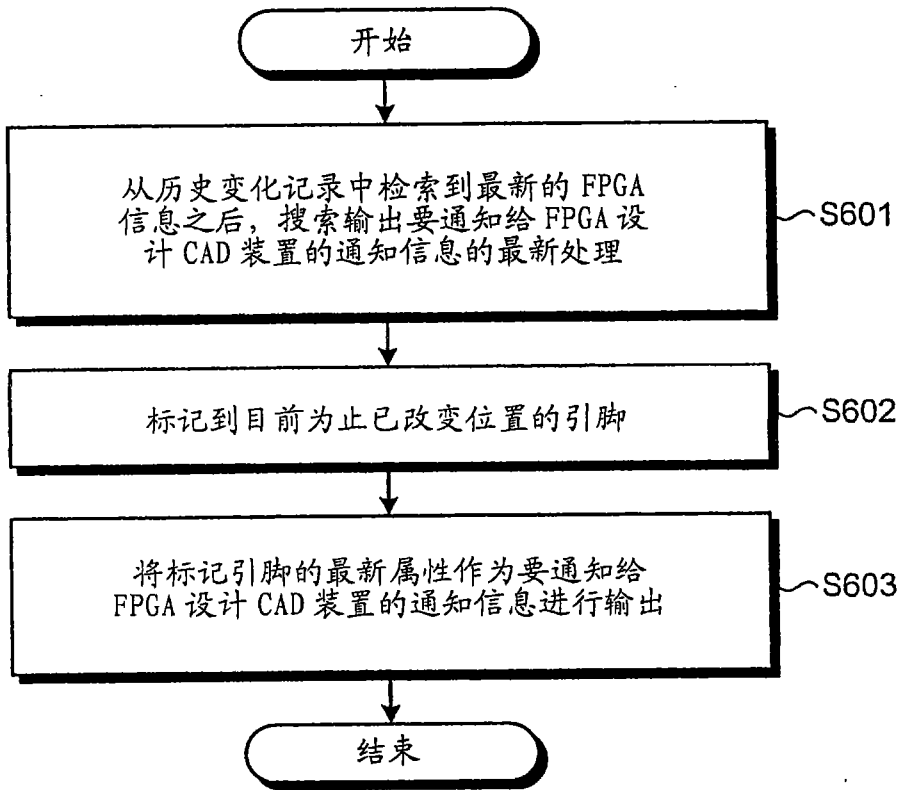


图 18

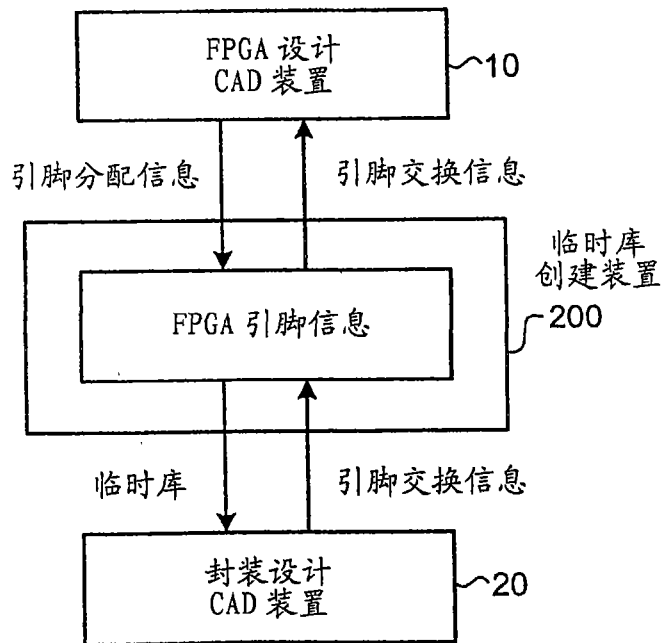


图 19

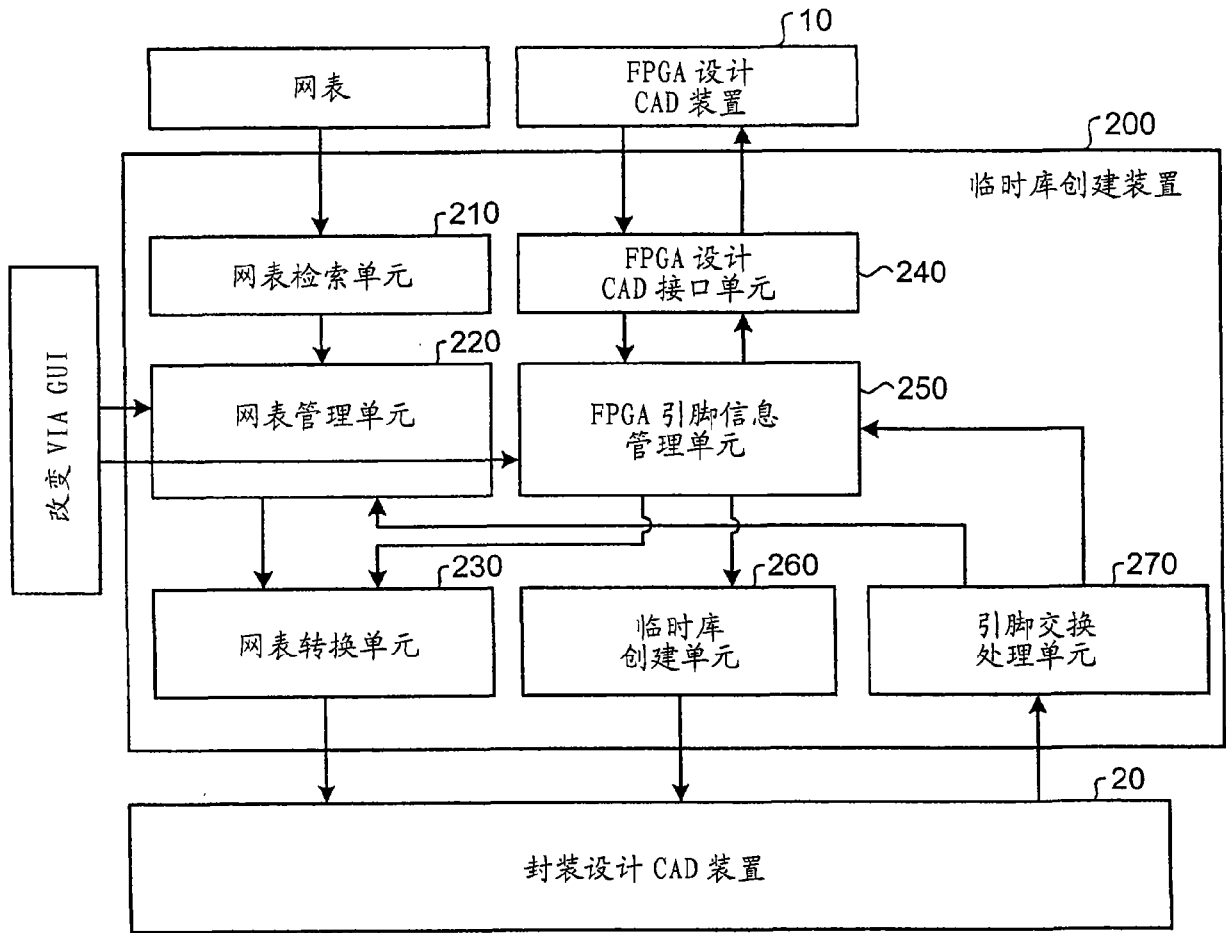


图 20

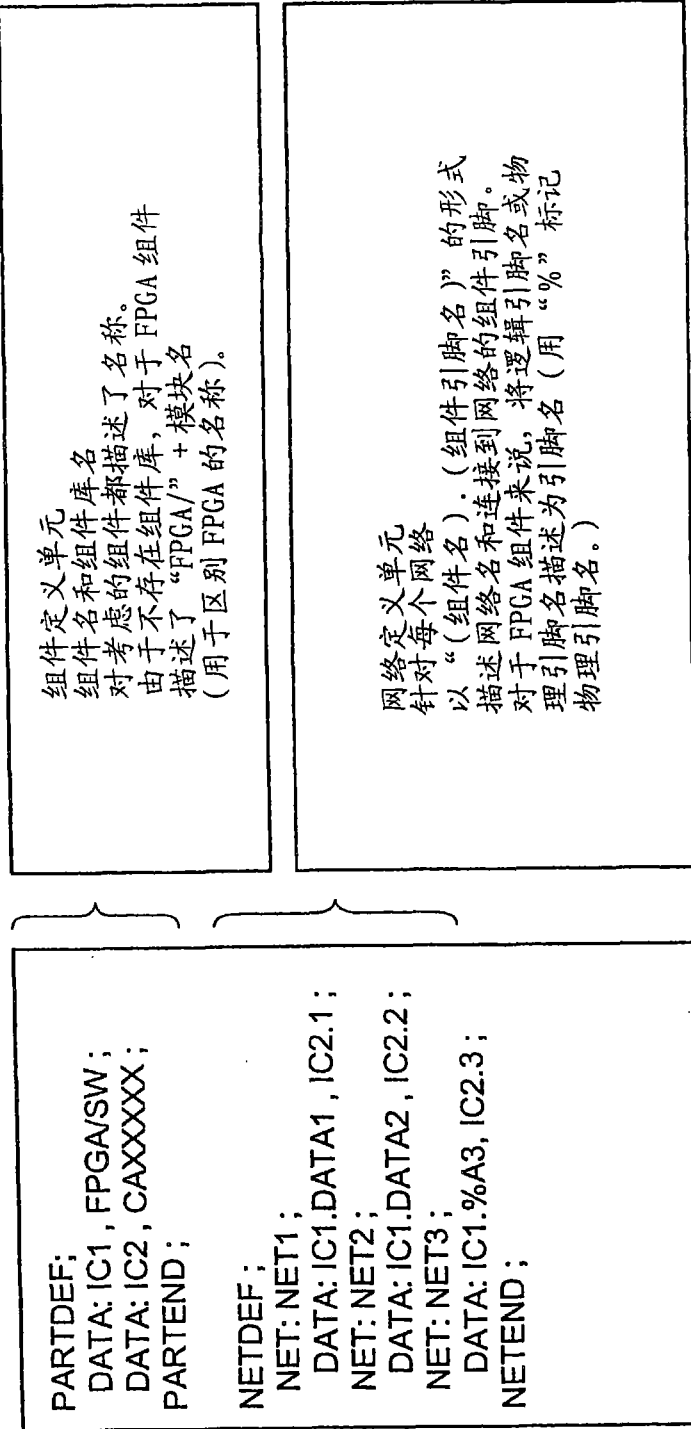


图 21

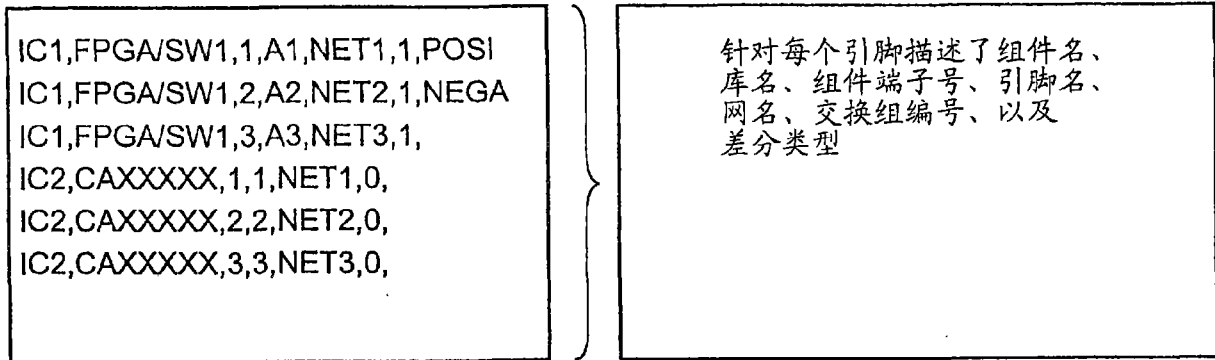


图 22

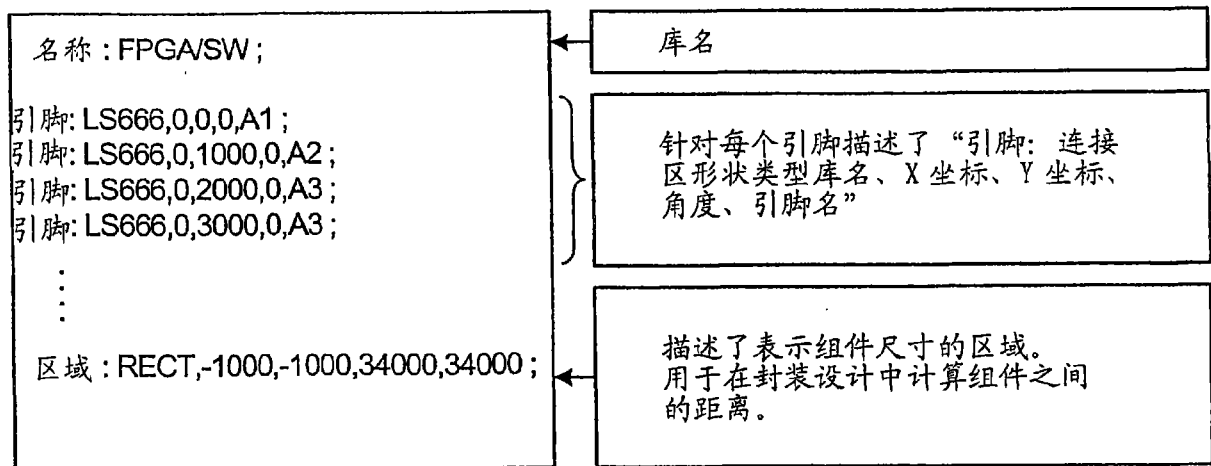


图 23

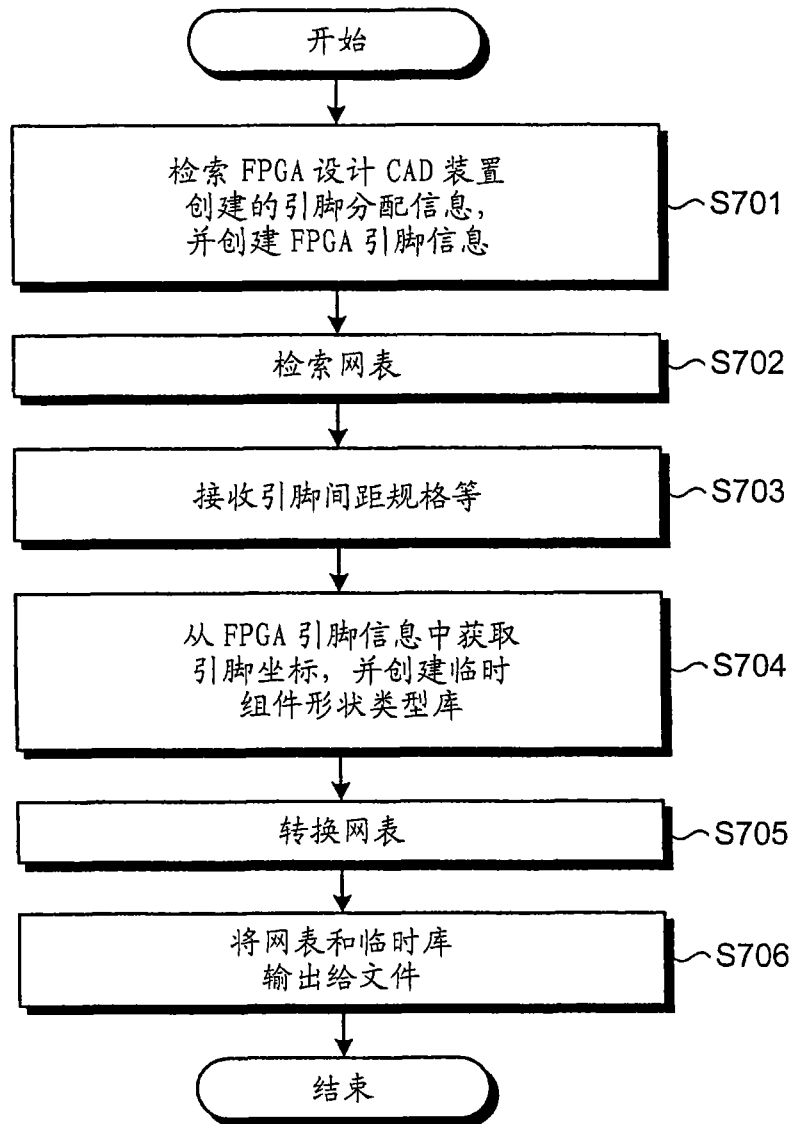


图 24

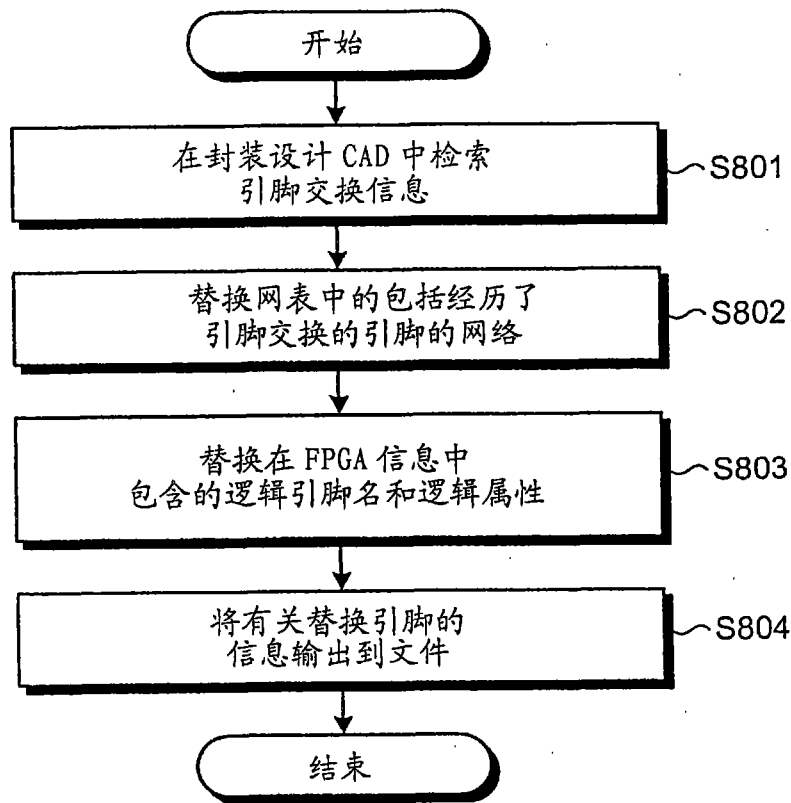


图 25

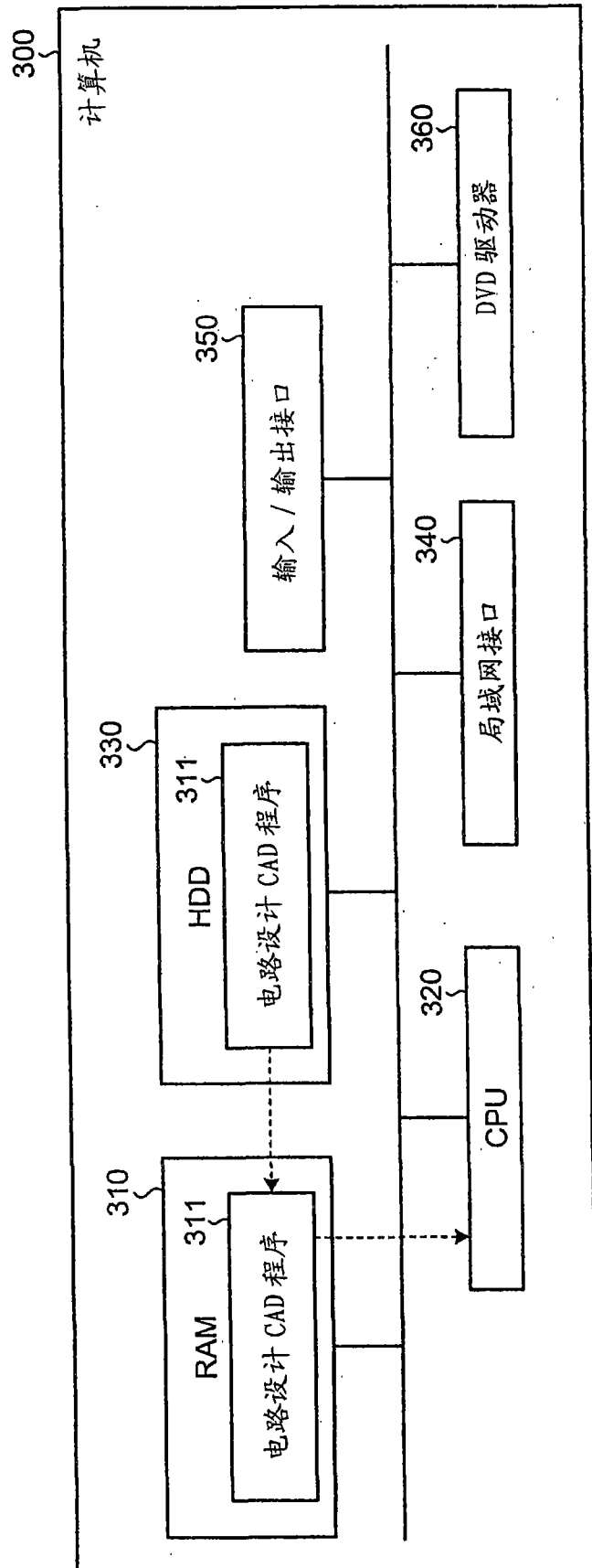


图 26