

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/50 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680034959.0

[43] 公开日 2009 年 1 月 28 日

[11] 公开号 CN 101356531A

[22] 申请日 2006.8.11

[21] 申请号 200680034959.0

[30] 优先权

[32] 2005.8.12 [33] GB [31] 0516634.3

[86] 国际申请 PCT/GB2006/002994 2006.8.11

[87] 国际公布 WO2007/020391 英 2007.2.22

[85] 进入国家阶段日期 2008.3.21

[71] 申请人 苏塞克斯大学

地址 英国布莱顿市

[72] 发明人 A·D·汤普森

[74] 专利代理机构 北京戈程知识产权代理有限公司
代理人 程伟

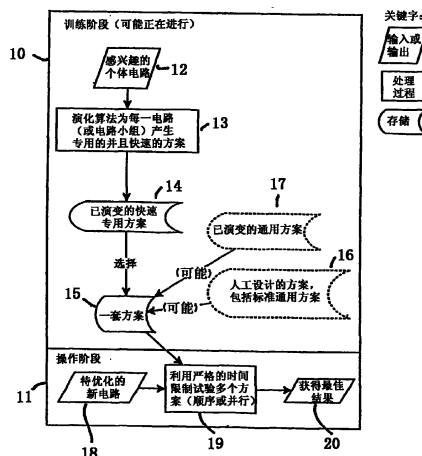
权利要求书 9 页 说明书 38 页 附图 5 页

[54] 发明名称

电子电路设计

[57] 摘要

一种用于优化待设计的电子电路的系统具有两个部分或阶段，其中用于使用演化算法来推导所选电子电路的优化方案的训练阶段(10)和其中用于使用推导出的优化方案来优化待设计的新电子电路的操作阶段(11)。所述训练阶段(10)使用演化算法来为多个输入电路或电路组的每一个生成专用和相对快速的优化方案(步骤13)。然后选择演化的一个或多个专用优化方案来形成一套优化方案(步骤15)，以使用来优化待设计的新电路。在操作阶段(11)中，在步骤(18)输入待优化的新电路，然后使用出自该套优化方案(15)的多个优化方案来试图优化新电路(步骤19)。把最佳优化结果视作对所述电路的优化(步骤20)。



1、一种用于生成用来自动设计电子电路的一套优化方案的方法，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；以及

把为不同电路演变的一个或多个优化方案包括在用来在其设计期间优化电子电路的一套优化方案中。

2、如权利要求1所述的方法，包括为从已知的基准电路中取出的一个或多个电路演变优化方案。

3、如权利要求1或2所述的方法，包括为一套多个个体电路演变单一的优化方案。

4、如权利要求1、2或3所述的方法，包括为相同的电路或电路组演变多个优化方案。

5、如前述任一项权利要求所述的方法，包括将方案优化的速度设置为演化算法的演变准则之一。

6、如前述任一项权利要求所述的方法，包括将所述方案的存储器使用需求设置为演化算法的演变准则之一。

7、如前述任一项权利要求所述的方法，包括评估已演变的优化方案，并且根据所述评估把它们中的一个或多个包括在一套优化方案中以供使用。

8、如前述任一项权利要求所述的方法，包括：

评估已演变的优化方案是否能够用来优化不同于已被演变的电路的一个或多个电子电路；以及

根据所述评估把优化方案包括在或者不包括在一套优化方案中来

使用。

9、一种用于选择优化方案以便包括在一套优化方案中以供在设计电子电路时使用的方法，包括：

为所选的目标电子电路推导优化方案；

评估所推导的优化方案是否能够用来优化不同于其目标电路的电子电路；以及

根据这种评估来选择是否把所述优化方案包括在一套优化方案中以供设计电子电路时使用。

10、如前述任一项权利要求所述的方法，包括经由演化运算从部分方式中选择包括在一套优化方案中的一个或多个优化方案。

11、如前述任一项权利要求所述的方法，包括把处于一套优化方案中的优化方案与一个或多个电路或电路类型相关联。

12、如前述任一项权利要求所述的方法，还包括：

使用来自该套多个优化方案中的两个或更多优化方案对待优化的电子电路的设计方面进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供电路设计方面使用。

13、一种用于优化电子电路设计的方法，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同电路演变的两个或更多优化方案包括在一套优化方案中以便在它们的设计期间用来优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案来对待优化的电子电路的设计方面进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供电路设计方面使用。

14、如权利要求 12 或 13 所述的方法，包括为每一优化方案设置时间限制以便完成其优化，并且根据达到时间限制时或优化已经结束时哪个更快，将达到时间限制时或优化已经结束时的优化结果作为方案的优化结果。

15、如权利要求 12、13 或 14 所述的方法，包括在方案的每一优化步骤之后确定优化的质量，并且把最佳确定结果作为优化方案的结果。

16、如权利要求 12、13、14 或 15 所述的方法，包括：
使用已经为特殊的目标电路推导出的优化方案来为不同于目标电路的电路进行优化。

17、一种用于优化待设计的电子电路的方法，所述方法包括：
使用已经为特殊的目标电路推导出的优化方案来优化不同于所述目标电路的电路。

18、如权利要求 16 或 17 所述的方法，包括：
当优化不是目标电路的电路时，依照不同于已经为目标电路推导出的优化方案的执行方式的方式来执行优化方案。

19、一种用于推导在设计电子电路时使用的优化方案的方法，包括：
使用演化算法来推导优化方案以便在设计电子电路方面使用，其中所述演化算法的准则是使用中所述优化方案优化电路设计方面将采用的速度。

20、一种用于推导在设计电子电路时使用的优化方案的方法，包括：
使用演化算法来推导优化方案以便在设计电子电路方面使用，其中所述演化算法的准则是使用中当优化电路设计方面时所述优化方案将使用的存储器资源。

21、一种用于生成在自动设计电子电路时使用的一套优化方案的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案的装置；以及

用于在其设计期间提供为不同电路演变的一个或多个优化方案作为用来优化电子电路的一套优化方案的装置。

22、如权利要求 21 所述的设备，包括用于为一套多个个体电路演变单一的优化方案的装置。

23、如权利要求 21 或 22 所述的设备，包括用于为相同的电路或电路组演变多个优化方案的装置。

24、如权利要求 21 至 23 中任一项所述的设备，包括将用于方案优化的速度设置为演化算法的演变准则之一的装置。

25、如权利要求 21 至 24 中任一项所述的设备，包括将用于方案的存储器使用需求设置为演化算法的演变准则之一的装置。

26、如权利要求 21 至 25 中任一项所述的设备，包括用于评估已演变的优化方案并且根据评估把它们中的一个或多个包括在一套优化方案中以便使用的装置。

27、如权利要求 21 至 26 中任一项所述的设备，包括：

评估已演变的优化方案是否能够用来优化不同于已被演变的电路的一个或多个电子电路；以及

根据所述评估把优化方案包括在或者不包括在一套优化方案中以便使用。

28、一种用于选择优化方案以便包括在一套优化方案中以便在设计电子电路时使用的设备，包括：

用于为所选的目标电子电路推导优化方案的装置；

用于评估所推导的优化方案是否可以用来优化不同于其目标电路的电子电路的装置；以及

用于根据这种评估来选择是否把所述优化方案包括在一套优化方案中以供设计电子电路使用的装置。

29、如权利要求 21 至 28 中任一项所述的设备，包括用于经由演化运算从部分方式中选择包括在一套优化方案中的一个或多个优化方案的装置。

30、如权利要求 21 至 29 中任一项所述的设备，包括用于把处于一套优化方案中的优化方案与一个或多个电路或电路类型相关联的装置。

31、如权利要求 21 至 30 中任一项所述的设备，还包括：

用于使用来自该套多个优化方案中的两个或更多优化方案来对待优化的电子电路的设计方面进行优化的装置；以及

用于选择根据多个优化方案确定的优化结果之一作为优化方案以供电路设计方面使用的装置。

32、一种用于优化电子电路设计的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案的装置；

用于提供为不同电路演变的两个或更多优化方案作为一套优化方案以用来在其设计期间优化电子电路的装置；

用于使用来自一套优化方案的两个或更多优化方案对待优化的电子电路进行优化的装置；以及

用于选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用的装置。

33、如权利要求 31 或 32 所述的设备，包括用于为每一优化方案

设置时间限制以便完成其优化的装置，以及用于根据达到时间限制时和优化已经结束时哪个更快，将达到时间限制时或优化已经结束时的优化结果作为方案的优化结果的装置。

34、如权利要求 31、32 或 33 所述的设备，包括用于在方案的每一优化步骤之后确定优化质量的装置，以及用于把最佳确定的结果作为优化方案的结果的装置。

35、如权利要求 31、32、33 或 34 所述的设备，包括用于使用已经为特殊目标电路推导出的优化方案来为不同于目标电路的电路进行优化的装置。

36、一种用于优化待设计的电子电路的设备，所述设备包括：

用于使用已经为特殊目标电路推导出的优化方案来优化不同于目标电路的电路的装置。

37、如权利要求 35 或 36 所述的设备，包括：

用于当优化不是目标电路的电路时，依照不同于已经为目标电路推导出的优化方案的执行方式的方式来执行优化方案的装置。

38、一种用于推导在设计电子电路时使用的优化方案的设备，包括：

用于使用演化算法来推导优化方案以便在设计电子电路方面使用的装置，其中所述演化算法的准则是使用中所述优化方案优化电路设计方面将采用的速度。

39、一种用于推导在设计电子电路时使用的优化方案的设备，包括：

用于使用演化算法来推导优化方案以便在设计电子电路方面使用的装置，其中所述演化算法的准则是使用中当优化电路设计方面时所述优化方案将使用的存储器资源。

40、一套用来在电子电路设计期间优化电子电路的优化方案，包括：

已经通过使用一个或多个演化算法推导出来的多个优化方案，用于为多个不同的电子电路的每一个演变优化方案。

41、一种电子电路，所述电子电路通过如下方式来优化，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同电路演变的两个或更多优化方案包括在一套优化方案中，以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用。

42、一种使用如权利要求 1 至 20 中任一项所述的方和法/或使用如权利要求 21 至 39 中任一项所述的设备来设计的电子电路。

43、一种用于构造电子电路的方法，包括：当构造电子电路时，使用如权利要求 1 至 20 中任一项所述的方法。

44、一种用于构造电子电路的设备，包括如权利要求 21 至 39 中任一项所述的设备；以及用于构造电子电路的装置。

45、一种用于构造电子电路的方法，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同电路演变的两个或更多优化方案包括在一套优化方案中以便在其设计期间用来优化电子电路；

使用来自该套多个优化方案中的两个或更多优化方案来对待优化的电子电路进行优化；

选择根据多个优化方案确定的优化结果之一作为优化方案以供电路使用；以及

使用所选择的电路优化方案来设计并构造电子电路。

46、一种用于构造电子电路的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案的装置；

用于提供为不同电路演变的两个或更多优化方案作为一套优化方案以用来在其设计期间优化电子电路的装置；

用于使用来自一套优化方案的两个或更多优化方案对待优化的电子电路进行优化的装置；

用于选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用的装置；以及

用于使用所选择的电路优化方案来设计并且构造电子电路的装置。

47、一种电子电路，所述电子电路通过如下方式来构造，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同电路演变的两个或更多优化方案包括在一套优化方案中，以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用；以及

使用所选择的电路优化方案来设计并且构造电子电路。

48、一种计算机程序元件，包括计算机软件代码部分，当所述程序元件在数据处理装置上运行时，用于执行如权利要求 1 至 20、43 或 45 中任一项所述的方法。

49、一种基本如上面参考任何一个附图所描述的用于生成一套优化方案以供自动设计电子电路时使用的方法或设备。

50、一种基本如上面参考任何一个附图所描述的用于选择优化方案以便包括在一套优化方案中以供在设计电子电路时使用的方法或设备。

51、一种基本如上面参考任何一个附图所描述的用于优化电子电路设计或待设计的电子电路的方法或设备。

52、一种基本如上面参考任何一个附图所描述的用于推导优化方案以供设计电子电路时使用的方法或设备。

53、一种基本如上面参考任何一个附图描述的用来在其设计期间优化电子电路的一套优化方案。

54、一种基本如上面参考任何一个附图描述的电子电路。

55、一种基本如上面参考任何一个附图所描述的用于构造电子电路的方法或设备。

电子电路设计

技术领域

本发明涉及电子电路的设计，尤其涉及（但并不限于）对数字电子电路的优化。

背景技术

电子电路的设计经常通过使用例如用于自动化且帮助某些或所有设计过程的软件、工具来予以辅助。图 1 用示意图示出了这种“电子设计自动化”（electronic design automation, EDA）过程的示例性步骤。对于在图 1 中所示出的每一步骤，将使用例如软件工具来辅助或执行设计过程。

在诸如图 1 所示的设计过程中，来自用户的主要的、最初的输入是对所期望的电路的高级说明（步骤 1）。此说明可以作为示意图（电路图）来阐明或者使用硬件描述语言来更典型地描述。所述高级说明例如还可以涉及预先设计的电路或子系统。

设计过程中的下一阶段就是所谓的高级合成（步骤 2）。该阶段把所输入的高级说明变换为几乎能够被映射到电子组件网络上的形式。此高级合成例如包括向特殊的电路模块分配任务以及为将如何处理这些任务安排进度。

然后是通常统称为“低级”合成 8 的一系列过程。这种低级合成把高级合成的结果变换为能够使用所选的制造工艺被构建的形式（通常是某种集成电路的形式）。

这种低级合成过程首先是与技术无关的优化（步骤 3）。此过程旨在尽可能地简化设计，但仍然处于抽象逻辑级别，而不是所述技术的物理组件的网络。典型情况下，与技术无关的优化在布尔方程的节点处操纵有向无环图（directed acyclic graphs, DAG）。对此逻辑网络的简化通常导致更小的物理电路。

下一阶段是技术映射（步骤 4），该阶段把所述设计映射到所选的

制造工艺中可利用的组件上。然后是与技术有关的优化步骤（步骤 5），该步骤试图执行进一步的优化。这些步骤都使用组件库 6。

然后，最后阶段是自动配置并且路由组件的步骤（步骤 7），该步骤试图找到所述组件的好的物理位置和用于在它们之间连接的路由。

正如本领域技术人员将理解的，以上是对电子电路设计流程的简化描述，并且由此存在且能够进行对所描述过程的变化和修改。例如，也可以是这样的情形，在如图 1 所示的步骤之间没有这样清楚的划分，并且所述过程还能够并且将返回以便重新考虑早期在更高级别所做的决定。除所示出的那些以外，还可能要（并且实际上一般将会）考虑的诸如有能够在其制造和应用过程中充分测试最终电路的设计验证以及容易程度（其“可测试性”）。

此种电子电路设计过程的关键方面是在设计流程中的许多阶段进行的自动化优化过程。例如，这种优化一般涉及使最终电路的尺寸最小（因为这将会减少成本和电路的硅覆盖区），而且还可以或作为替代，一般涉及例如电路操作的速度、电路的功率消耗、电路的可测试性等等。

用于与技术无关的优化的优化准则的一个例子是使表示所述电路的布尔函数的有向无环图中因子形式的文字之和最小。此准则提供了对表示所述电路的布尔函数的有向无环图中所述逻辑的整体复杂度的测量，并且使它最小化能够产生具有可接受的延迟特性的更小的物理电路。通过此准则的优化经常被认为是有用的，即便以后执行其它更专用的优化。用于优化此准则的优化过程将通常指定一个用于使表示所述电路的布尔函数的有向无环图中因子形式的文字之和最小的过程。

在电子设备设计自动化工具中使用的优化过程能够包括许多步骤，而排列这些步骤的顺序和设置它们的参数是一个难题。待执行的优化步骤的说明通常被称为优化方案或脚本。可以把优化方案看作指定多个变换过程，所述变换过程依照期望的优化准则共同改进电子电路设计的质量。优化脚本是优化方案的一个例子（形式）。然后，所述优化方案（例如脚本）可以被内置于软件设计工具，并且在电路设计过程期间用作优化过程。

通常使用的并且众所周知的电子设计自动化工具是由位于 Berkeley 的加州大学开发的 Berkeley SIS 系统。在 SIS 系统中，优化方案被配置为脚本，该脚本阐明优化过程的步骤且作为文本文件提供给 SIS 软件。

一种用来推导用于电路设计的优化方案的已知方式是人工分析和评估不同的优化方案。这导致了許多已知的、标准优化方案（脚本）的开发，人们已经发现这些方案可以较好地优化几乎所有的电路。这些优化方案可以被称作“通用”方案或脚本。例如，已经为 SIS 系统开发了很多这种通用优化方案。

试图推导更专用的优化方案也是已知的，例如适合优化一个特殊电路的优化方案（并且可能会超过更通用的方案）。这通常从已知的通用方案开始，通过人工实验来完成。

此外，已经有人建议使用图 2 所示类型的所谓演化（或遗传）搜索算法来测试和优化优化方案。已经发现，可以使用这种演化算法来推导优化方案，所述优化方案与那些已经由行业专家设计的更通用的优化方案相比，对于给定目标电路而言，将会完成的更好。

在使用演化算法来生成给定电路的优化方案的布置中，正如本领域技术人员将理解的，在可以将其用作优化上述电路的工具之前，必须“演化”所述优化方案。因此，可以把整个优化过程视为具有两个部分，第一是“训练”阶段，其中使用所述演化算法来生成上述电路的优化方案，第二是“操作”阶段，其中把由训练阶段生成的演化的优化方案应用于待优化的电路以便优化该电路。换言之，首先使用所述演化算法来生成优化方案，然后把演化的方案用作用来优化待优化的电路的工具。

利用更专用的优化方案的演化推导的缺点在于：这种演化推导是一个时间和计算资源密集的过程（因为所述过程包括评估许多不同的潜在候选优化方案）。在待优化的电路具有特殊的重要性或影响的情况下，可以证明这种努力是正确的，但这意味着对于为可能遇到的每一个电路试图演化专用的优化方案而言，实际上并不可行。

由于需要更大的计算工作量和更多时间，并且由于在通常情况下要超越人工推导的方案困难，所以通常也不使用演化算法来产生将

优化大量电路的通用方案。

这都意味着实际上，电路设计系统趋向于使用更通用、人工推导的方案作为它们的优化方案。

发明内容

申请人认为对于用于设计电子电路的自动工具和过程还有改进的范围。

依照本发明的第一方面，提供了一种用于生成用来自动设计电子电路的一套优化方案的方法，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演化优化方案；以及

在它们的设计期间，把为不同电路演化的一个或多个优化方案包括在用来优化电子电路的一套优化方案中。

依照本发明的第二方面，提供了一种用于生成用来自动设计电子电路的一套优化方案的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演化优化方案的装置；以及

用于在它们的设计期间把为不同电路演化的一个或多个优化方案作为用来优化电子电路的一套优化方案的装置。

依照本发明的第三方面，提供了一套用来在电子电路设计期间优化电子电路的优化方案，包括：

通过使用一个或多个演化算法来为多个不同电子电路的每一个演化优化方案而推导出来的多个优化方案。

在本发明中，使用演化算法来为很多不同的电路推导优化方案。然后提供这样演化的优化方案作为一套（一组）优化方案，以用来优化待设计的新电路。

申请人已经发现，依照这种方式推导一组演化的优化方案能够提供一组优化方案，其在电路设计期间，与例如已知的、更通用的优化方案相比，例如将提供对新电路的更好的优化，而不需要例如使用演化算法来为将会或可能会预料遇到的每一个电路推导优化方案。

具体地，申请人已经发现，为给定电路专门演化的优化方案（专

用方案) 对于其目标电路而言不仅完成的更好, 而且对于某些其它电路而言也将趋向于完成的更好。可以把这些其它电路看作是所述优化方案的“辅助”电路。申请人已经进一步认识到, 通过开发一套多个专用优化方案, 每一个均具有其自己辅助电路组, 那么所述专用方案的组合连同其辅助电路组一起可以提供一组优化方案, 该组优化方案能够并且将会覆盖大多数可能遇到的电路, 但不一定都能覆盖, 并且不需要为可能遇到的每一个单个电路演化专用优化方案。实际上, 申请人已经发现: 对许多电路只使用一套少数专用优化方案也能够获得优良的性能。

能够如所期望地选择为其演化优化方案的电路。例如, 能够为从通常用在电子设备设计自动化工具中的已知、参考或基准组电路中取出的一个或多个(所选) 电路演化方案。然而, 这并不必要, 例如也能够是非基准的其它电路并且还优选使用它或者作为替代来使用。例如, 能够为一个或多个新的、未知的电路, 例如是特别感兴趣的电路演化优化方案。

例如, 能够随机选择为其演化优化方案的电路。然而, 在优选实施例中, 可以为人们认为对已知、标准方案难以优化的电路演化方案。优选地为较难优化的电路演化优选方案。例如可以(并且实际上是优选) 根据例如现有的一套优化方案(以及例如, 在该套中任何已标识的缺点) 来选择所述电路。

在优选实施例中, 选择待评估的多个不同电子电路(即, 将为其演化优化方案)。在一个优选实施例中, 为 5 到 15 个电路演化优化方案, 最优选的是 10 个电路。

在优选实施例中, 可以把多个单个电路组合在一起, 并且为该组电路演化单个优化方案。例如, 这样会允许产生专用于特殊种类电路的优化方案。例如在特殊类型或种类的电路能够由一(小) 组单个电路表示的情况下, 这样是十分有用的, 该组单个电路在优化方案的演化过程中全都能够被测试。

例如, 为相同的电路或电路组演变多个优化方案也是可能的(并且例如在一套待使用的优化方案内包括所有那些优化方案也是可能的)。正如稍后将论述的, 申请人已经发现为相同电路演变的不同优化

方案仍会具有不同的辅助电路组，因此在待使用的一套优化方案（优化方案集）中包括两个或所有优化方案是十分有用的。

能够使用任何适当的演化算法或技术来演变每一电路（或电路组）的优化方案，诸如本领域中已经公知的演化技术。因此，例如，演化运算能够开始于一群随机产生的方案，或者能够以已经被人工演变或设计的方案为基础而逐渐成熟。相同的演化算法可用于每一个电路，或者也能够使用不同的算法。

正如本领域技术人员将理解的，所述演化过程应当把旨在优化的优化方案的优化准则作为目标（即，作为选择（适合）准则），诸如因子形式的文字数目总和。这样，例如能够确定每一候选方案的优化准则结果，然后对应地对所选的候选方案进行进一步的演化或否决，正如本领域中所公知的。

在特别优选的实施例中，除在演化过程期间根据它们如上所述的优化“结果”来评估候选优化方案之外，还要考虑所述方案到终止所花费的时间（即，它们的执行速度），越迅速的方案越是优选的（例如，优选连续的演化和/或选择作为优化方案来使用）。最优选地，如果对两个或多个候选方案的优化准则测量结果相同，那么优先选择更快终止的方案。

因此，在特别优选的实施例中，为演化算法设置并且其所支持的演变准则之一是方案优化的速度（即，方案将为其目标电路生成其优化结果（即，优化）的迅速程度）。这样不仅能够更好地演变出对它们的电路能生成相对高质量的结果的优化方案，而且相对更快。在使用一套优化方案的过程中，这样是十分有益的，正如稍后将论述的。

在优选的配置中，为优化方案生成其优化结果（即，直到终止）所花费的时间长短设置了时间限制，例如，超过此时间限制的任何方案在到达所述时间限制时被终止（同时把由此获得的优化结果作为方案的结果），或者例如根据进一步的考虑而被否决。除上述快速终止方案的优先选择之外此时间限制是优选的。优选地，此操作至少在优化方案演化的早期阶段期间完成。在演化过程的后期，所述时间限制例如还可以被增加，而非完全消除。此时间限制例如能够以采取已知的、通用脚本获得所讨论的电路的结果需要多久为基础。适当的这种时间

限制例如可以是 600 秒或更少。

限制优化方案开始执行的时间还使演化过程本身变得非常便利，因为它有助于确保优化方案能够在待演变方案数目合理的情况下被足够快速地演变，并且有助于确保在演化过程期间所述优化方案在合理的时间内被评估。

可以认为，包括优化速度作为演化推导电子电路设计的优化方案的准则是新颖的，并且在其自身权利方面是有益的。

因此，依照本发明的第四方面，提供了一种用于推导在设计电子电路时使用的优化方案的方法，包括：

使用演化算法来推导在设计电子电路方面使用的优化方案，其中所述演化算法的准则是使用中所述优化方案优化电路设计方面将采用的速度。

依照本发明的第五方面，提供了一种用于推导在设计电子电路时使用的优化方案的设备，包括：

用于使用演化算法来推导在设计电子电路方面使用的优化方案的装置，其中所述演化算法的准则是使用中所述优化方案优化电路设计方面将采用的速度。

在特别优选的实施例中，除如上所述那样，在演化过程期间根据它们的优化“结果”以及例如它们到终止所花费的时间来评估候选优化方案之外，还要考虑在其执行期间所述方案的存储器使用量，此时使用（“消耗”）越少存储器的方案越是优选的（例如，优先选择连续的演化和/或选择作为优化方案来使用）。最优选地，如果对两个或多个候选方案的优化准则测量结果相同，那么优先选择更低存储器使用量的方案。

因此，在特别优选的实施例中，为演化算法设置并且为其所支持的演变准则之一是方案的存储器使用需求（例如，方案将使用或需要为其目标电路生成其优化结果（即，优化）的存储器资源有多少）。这样不仅能够更好地演变出对它们的电路能生成相对高质量的结果的优化方案，而且将只需相对更少的存储器使用量。在使用一套优化方案时，这也是十分有益的。

可以认为，包括存储器使用量作为演化推导电子电路设计的优化

方案的准则是新颖的，并且在其自身权利方面是有益的。

因此，依照本发明的第六方面，提供了一种用于推导在设计电子电路时使用的优化方案的方法，包括：

使用演化算法来推导在设计电子电路方面使用的优化方案，其中所述演化算法的准则是使用中在优化电路设计方面时、所述优化方案将使用的存储器资源。

依照本发明的第七方面，提供了一种用于推导在设计电子电路时使用的优化方案的设备，包括：

用于使用演化算法来推导在设计电子电路方面使用的优化方案的装置，其中所述演化算法的准则是使用中在优化电路设计方面时、所述优化方案将使用的存储器资源。

对于演化算法而言更优选的是，优先选择（用于进一步演化或作为优化方案来使用）更短的方案（例如，在所述方案的优化质量和执行时间相等时）。

在优选的实施例中，能够获得并且演变跨越或包括多个优化准则或过程的优化方案（例如，通常可以被独立考虑）。例如，当演变优化方案时还可以考虑后续技术映射，而不是孤立地执行与技术无关的优化。如果这种“扩展”优化评估不产生演化过程并且评估的完成费时太久，那么最好是只执行它。

在特别优选的实施例中，所述一个或多个演化算法被设置并且选择，使得它们将在可以接受的短时间段内为特殊电路演变优化方案。这样将允许所述演化过程在可以接受的短时间量程中重复几次。

在优选的实施例中，允许演化算法演变长的方案，所述方案可能包含重复的部分，但是并非限制要这样做。另外优选的是，从已演变的优化方案（在演化之后）中删除（删节）任何冗余的命令，例如可通过使用自动化的测试体系集来查看哪些命令实际上是必要的。在特别优选的实施例中，可以把方案的“删节”和“未删节”版本都包括在一套优化方案中，因为它们例如可能具有不同的辅助电路组。

在优选的实施例中，例如通过执行并行运行在独立微处理器上的多个演化运算来并行演变一个或多个电路的优化方案。

能够如所期望地选择待使用的包括在一套优化方案中的已演变的

优化方案。把已演变的优化方案的各个以及每一个都包括在一套优化方案中以供使用，或者把并非全部优化方案包括在一套优化方案中，这都是可能的。在优选的实施例，把已演变的优化方案的选择数目包括在一套优化方案中，所述数目最好是两个或更多方案，最好是 10 个方案。

在优选的实施例，评估已演变的优化方案以便包括在一套优化方案中以供使用，并且是否包括在该套中取决于所述评估。这种评估能够依照任何适当的或期望的方式来进行。例如，可以使用已演变的方案来优化采样电路的选择，以便查看其包含物是否将增强该套优化方案，和/或查看其性能是否能够比得上标准人工设计的方案。在优选的实施例，评估所选择的方案测试集中的每一方案的性能，并且用于从测试集中选择最小数目的方案，这些方案将会提供期望的优化性能，以用作该套优化方案。

在特别优选的实施例，根据还可以有效用于优化的辅助电路（即，不同于其能够被有效用于优化的目标电路的电路），评估已演变的优化方案以便包含在一套优化方案中。为此，在这方面优选的是，通过相对于采样电路的（最优选地预定的）选择对方案进行测试，以此来评估和估计方案的辅助电路的数量和/或类型，而非试图确定方案的辅助电路的整个范围。

例如，可以考虑使用优化方案的辅助电路的数目，和/或能够将给定方案的辅助电路与其它一个或多个优化方案的辅助电路（例如已经包括在待用的一套优化方案中的方案）进行比较（例如，用于查看新的优化方案是否将被有效添加到该套优化方案）。

因而，在优选的实施例，已演变的优化方案的辅助电路组被评估（例如被估计），并且根据所述评估来确定是否把所述优化方案包括在一套优化方案中以供使用。

可以认为，这种配置是新颖的，并且在其自身权利方面十分有益。由此，依照本发明的第八方面，提供了一种用于选择包括在方法优化方案中以供在设计电子电路时使用的优化方案的方法，包括：

为所选的目标电子电路推导优化方案；

评估所推导的优化方案是否可以用来优化不同于其目标电路的电

子电路；以及

根据这种评估来选择是否把所述优化方案包括在一套优化方案中
以供设计电子电路使用。

根据本发明的第九方面，提供了一种用于选择包括在一套优化方案中
以供在设计电子电路时使用的优化方案的设备，包括：

用于为所选的目标电子电路推导优化方案的装置；

用于评估所推导的优化方案是否可以用来优化不同于其目标电路
的电子电路的装置；以及

用于根据这种评估来选择是否把所述优化方案包括在一套优化方案中
以供设计电子电路使用的装置。

正如本领域技术人员将理解的，如果适当，本发明的上述方面可以
包括此处所描述的本发明的所有优选和可选特征中的任意一个或更多个。
因而，例如所述优化方案最好使用演化算法来推导。

在本发明的这些方面和配置中，所述“辅助”电路还可以依照任何
期望的方式来确定和评估，其中为特殊的目标电路推导出的优化方案
也将会有效地优化所述辅助电路。例如，可以评估特殊的（诸如所选的）
电路集（诸如所选的基准或参考集中的每个电路）的优化方案的优化性，
并且如果一个电路的优化方案的优化性能好于该电路的已知通用优化方案
的优化性能，那么所讨论的电路可以被当作优化方案的辅助电路（因为
将会为该电路提供改进的优化性能）。

当确定是否将其包括在一套优化方案中时，除评估优化方案的“辅助”
电路之外，还优选或者可作为替代的是，基于优化方案的执行速度（如
上所述）把优化方案包括（或者不包括）在一套优化方案中，此时，
例如越快速的方案和/或终止得比所选的优选预先确定的时间限制更快
的那些方案，越优先被包括在一套优化方案中以供使用。

因而，正如根据上文可以理解的，优选地，本发明包含例如根据
将由每一优化方案优化的“辅助”电路来选择一个或多个已演变的优化
方案以便包括在一套优化方案中以供使用的步骤和装置。例如，这种
选择通常意味着并非所有已演变的优化方案被包括在一套优化方案中
以供使用，但是对于这种选择而言，使所有已演变的优化方案被使用
也仍然是可能的。

在优选的实施例中，能够经由演化运算从部分方式选择一个或多个优化方案包括在一套优化方案中，包含或者不是演化运算最终的已演变的优化方案。申请人已经发现：与完全为其目标电路磨练而成的最终结果相比，经由演化运算从部分方式选择的优化方案会具有不同并且实际上更有用的辅助电路组。

在优选的实施例中，除为指定的所选目标电路演变的方案之外，一套优化方案能够并且最好包括其它优化方案。这种额外的优化方案例如能够包括标准的、预先确定的和/或人工推导的通用优化方案，和/或甚至演变而来的通用优化方案（如果可获得的话）。例如，把现有的、已知的、标准人工设计的通用方案包括在一套优化方案中可确保利用一套优化方案取得的优化质量应当不劣于独自使用标准的通用方案时可获得的优化质量。

在依照这种方式把额外的优化方案包括在一套优化方案中的情况下，如果额外的方案在用于优化给定电路时能够足够迅速地操作（即，它们的优化速度足够快速，例如，低于所选的时间限制），那么最好是只包括额外的方案。

在优选的实施例中，把处于待用的一套优化方案中的优化方案与被认为将会特别有效地用于优化的一个或多个电路或电路类型相关联。当优化新电路时，这样可以更有利于选择一个或多个优化方案来使用。

正如本领域技术人员将理解的，以上主要涉及待用的一套优化方案的推导，也就是上述的“训练阶段”。

一旦一套优化方案已经生成，就可如所期望地使用，并且例如可依照本领域中已知的任何适当方式来使用，以便当设计电子电路时（即，在电路设计过程的“操作阶段”）来优化它们。因此，例如当新电路设计的各方面或准则（与已经推导出的一套中的优化方案的一个或多个方面或准则带有某些关系或者对应性等等）将被优化时，能够把该套优化方案用于所述优化过程。

在特别优选的实施例中，可使用处于一套优化方案中的多个优化方案中的每个来优化新的电路设计，然后把所有已测试的优化方案的结果之一（所选的一个）作为优化结果以供新电路（即，优化的电路

设计)使用。换言之,为所述电路依次试验多个优化方案,并且例如选择最佳的结果。

申请人已经发现:例如与简单使用标准的通用优化方案相比,这种配置将易于为新的、未知的电路提供最佳优化结果,并且不需要为可能遇到的每一个电路推导专门的优化方案。此外,利用依照本发明的方式推导的一套优化方案,特别是如果它们被演变为提供相对快速的优化,那么对于依照这种方式测试多个优化方案是切实可行的,并且期望改进优化结果是可行的。

因此,在优选的实施例中,本发明还包括用于使用来自该套多个优化方案中的两个或更多优化方案来对待优化的电子电路的设计方面进行优化的步骤或装置,以及用于选择根据多个优化方案确定的优化结果之一作为优化方案以供电路设计方面使用的步骤或装置。

在这些配置中,例如,简单使用处于一套优化方案中的所有优化方案进行优化过程也是可能的,并且例如选择最佳的一个也是可能的。作为选择,更有限的或所选的优化方案集例如能够基于上述电路的性质或类型来试验。例如,在已经根据更可能有效用于优化的一个或多个电路的类型把优化方案进行分类的情况下,后一方法可能特别有用。

在多个优化方案已经被试验之后所使用或所选择的优化结果能够依照任何适当的并且是期望的方式来选择。例如,就优化上述电路的问题或方面而言,最好选择用于提供良好结果(并且最好是最佳结果)的方案。例如,即使例如所有可能的优化方案仍未被试验,也能够获得在特殊的、最好是预先确定的时间段取得的最佳结果。

例如,在将要考虑多个优化准则(例如尺寸、速度、电源使用等等)的情况下,则例如能够基于在不同需求之间已选的、例如预先确定的权衡或分等来选择优化结果。例如,在该套优化方案提供多个可接受的优化结果或选项的情况下,也能够在这不同的这种权衡之间进行选择。这样能够进一步便于对一个或多个给定电路的设计探查和优化。

因此,选择的优化结果最好是基于使用优化方案取得的优化质量的测量结果。最优选地,选择用于提供最佳质量的优化结果的优化方案。这种优化质量能够依照任何适当的和期望的方式来测量。

在特别优选的实施例中,为每一优化方案考虑到了最好预先确定

的时间限制，以便完成对所述电路的优化，与此同时，例如达到时间限制或优化已经结束哪一个更快出现，则把那时的优化结果作为该方案的优化结果（然后所述系统转移至下一待试验的优化方案）。这样有助于确保即使有多个优化方案正被试验，所述处理也被足够快速地进行。

对于时间限制的设置（如果有的话）例如能够取决于所花费的时间和优化性能之间的权衡、和/或取决于例如由已知的、标准通用脚本获得电路的最佳优化结果和上述优化准则所花费的时间。

所述优化方案能够依照与演变它们时所采用的完全相同的方式来使用并且应用于新的电路。然而，在优选的实施例中，可以并且最好以不同于演变它们时（即，在训练阶段期间）所使用或执行的方式来使用和/或执行优化方案，因为这样十分有益。

例如，为特殊目标电路演变的专门优化方案通常将在优化方案结束时给予其目标电路最高优化的版本。然而，对于不是特定目标电路的电路而言，在执行优化方案期间，在某个中间点所述优化方案可能生成其最佳结果。

因而，在特别优选的实施例中，在方案的每一优化步骤之后测量优化的质量，并且如果合适的话，获取并且使用最佳测量结果作为该优化方案的结果（而非例如简单地采取优化方案的最后结果）。

类似地，当使用所述优化方案优化新电路时，优选的是允许进行优化方案的迭代（重复）。（正如本领域中所公知的，优化方案的迭代是有益的，但是在训练阶段期间（即，当首先推导方案时）不必要这样，这是因为在该阶段，在优化方案的单个迭代中可以允许单个方案来调节优化步骤序列的重复）。

可以认为，这种配置是新颖的，并且在其自身权利方面十分有益。因而，根据本发明的第十方面，提供了一种用于优化电子电路设计的方法，包括：

为所选的目标电子电路推导优化方案；

使用所推导的优化方案来优化不是所选的目标电路的电路；以及

当优化不是所选的目标电路的电路时，依照不同于已经为所选的目标电路推导出的优化方案的执行方式的方式来执行所述优化方案。

根据本发明的第十一方面，提供了一种用于优化电子电路设计的设备，包括：

用于为所选的目标电子电路推导优化方案的装置；

用于使用所推导的优化方案优化不是所选的目标电路的电路的装置；以及

用于当优化不是所选的目标电路的电路时，依照不同于已经为所选的目标电路推导出的优化方案的执行方式的方式来执行所述优化方案的装置。

可以依照要求来配置用于试验和评估给定电路的多个优化方案的过程。例如，可以依照随机顺序来依次试验每一优化方案。

在一个优选实施例中，每一优化方案在待优化电路的相同的初始描述下操作。在这种情况下，每一优化方案实际上将独立于其它优化方案来运行，因此所有优化方案实际上最好是例如在并行操作的多个处理器上并行执行。

在另一优选实施例中，所述优化方案顺序（一个接一个）执行，最优选地为序列中的每一方案使用任何先前方案找到的最佳结果作为其开始点。在这种情况下，如果希望的话，还可以选择优化方案的执行（试验）次序。

也可以（例如，如果时间允许）沿上述线路对优化方案的并行和顺序执行都进行试验。

应当理解，当正在使用优化方案优化电路设计方面时，通常不将它们用于为其已经特别演变了优化方案的目标电路，而是将用于优化没有为其进行特别设计的电路。

可以认为，这样是新颖的，并且在其自身权利方面十分有益。因而，根据本发明的第十二方面，提供了一种用于优化待设计的电子电路的方法，所述方法包括：

使用已经为目标电路推导出来的优化方案来优化不同于所述目标电路的电路。

根据本发明的第十三方面，提供了一种用于优化待设计的电子电路的设备，所述设备包括：

用于使用已经为特殊的目标电路推导出来的优化方案来优化不同

于所述目标电路的电路的装置。

虽然上文已经独立描述了电路设计过程的“训练”和“操作”阶段，但正如本领域技术人员所理解的，除涉及并且致力于这些单个处理过程之外，本发明还涉及并且覆盖用于推导优化方案然后使用它们来优化电子电路设计的组合操作。

因而，根据本发明的第十四方面，提供了一种用于优化电子电路设计的方法，包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同的电路演变两个或更多优化方案包括在一套优化方案中以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用。

根据本发明的第十五方面，提供了一种用于优化电子电路设计的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案的装置；

用于提供为不同的电路演变两个或更多优化方案作为一套优化方案以用来在其设计期间优化电子电路的装置；

用于使用来自该套优化方案的两个或更多优化方案对待优化的电子电路进行优化的装置；以及

用于选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用的装置。

根据本发明的第十六方面，提供了一种已经通过如下方式得以优化的电子电路，所述方式包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同的电路演变两个或更多优化方案包括在一套优化方案中以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用。

正如本领域技术人员所理解的，如果适当，本发明的这些方面和实施例可以并且最好包括此处所描述的本发明的所有优选和可选特征中的任意一个或更多个。

本发明还因此涉及使用本发明的技术来构造电子电路并且涉及已经使用本发明的技术构造出来的电子电路。所述电路本身可以依照任何适当的方式来构造，例如通过使用已知的电路设计和制造技术来构造。

因而，根据本发明的第十七方面，提供了一种用于构造电子电路的方法，包括：

使用一个或多个演化算法为多个不同的电子电路的每一个演变优化方案；

把为不同的电路演变两个或更多优化方案包括在一套优化方案中以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用；以及

使用所选择的电路优化方案来设计并且构造电子电路。

根据本发明的第十八方面，提供了一种用于构造电子电路的设备，包括：

用于使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案的装置；

用于提供为不同的电子电路演变两个或更多优化方案作为一套优化方案以用来在其设计期间优化电子电路的装置；

用于使用来自该套优化方案的两个或更多优化方案对待优化的电子电路进行优化的装置；

用于选择根据多个优化方案确定的优化结果之一作为优化方案以

供所述电路使用的装置；以及

用于使用所选择的电路优化方案来设计并且构造电子电路的装置。

根据本发明的第十九方面，提供了一种电子电路，所述电子电路已经通过如下方式被构造，所述方式包括：

使用一个或多个演化算法来为多个不同的电子电路的每一个演变优化方案；

把为不同的电路演变的两个或更多优化方案包括在一套优化方案中以用来在其设计期间优化电子电路；

使用来自该套多个优化方案的两个或更多优化方案对待优化的电子电路进行优化；以及

选择根据多个优化方案确定的优化结果之一作为优化方案以供所述电路使用；以及

使用所选择的电路优化方案来设计并且构造电子电路。

正如本领域技术人员将理解的，如果适当，本发明的这些方面和实施例可以并且最好包括此处所描述的本发明的所有优选和可选特征中的任意一个或更多个。

本领域技术人员还将理解的是，所述训练和操作阶段可以一个接一个地实施，并且例如使用相同的硬件和/或软件来实施，或者可以被单独地承载并且位于不同位置，和/或由不同的个体和/或组织来实施。例如，可以由电子设计自动化工具销售商来推导优化方案，然后由 EDA 工具的客户或最终用户来执行电路优化（操作阶段）。同样地，个体或组织可以推导他们自己的一套优化方案和/或一套优化方案库，然后使用它们来优化电路设计。

还应理解的是，优化方案的演化以及把它们包括在一套优化方案中来使用可以并且最好是正在进行的过程。因而，一旦开始采用操作阶段，则不需要停止训练阶段。例如，由正在进行的训练阶段识别和推导的额外的有益优化方案可以被添加到该套优化方案以供使用。同样地，为其演变优化方案以便包括在一套优化方案中以供使用的新电路可以根据在使用该套优化方案优化电路设计期间（即，在操作阶段期间）识别出的微弱或不良优化性能来识别。

本发明可以用来推导优化方案并且把优化方案用于任何适当的电子设计自动化工具，例如本领域中早已公知的工具和技术。尽管并非一定如此，但本发明特别适合于采用 Berkeley SIS 系统的优化方案（脚本）使用并且特别适用于该系统。在把本发明应用于 Berkeley SIS 系统的过程中，不必修改 SIS 软件本身。

本发明可以应用于优化任何并且所有类型的电路设计，诸如通用处理器、数字信号处理器、专用信号处理器、现场可编程器件、专用集成电路、物理上优化的集成电路以及芯片级系统的集成电路。它不仅特别适用于数字电子线路，而且如果期望，也可以用于模拟电路。本发明还因此扩展到已经使用本发明的任何方法和/或设备演化来的电子电路，并且扩展到用于构造电子电路的设备或方法，包括使用本发明的任何方法和/或设备来构造电路本身的步骤或装置。

正如本领域技术人员将理解的，此处所描述的本发明的所有方面可以并且最好包括使用此处所描述的本发明的所有可选和优选特征中的一个或多个，如果适当。因而，例如在将要为电路推导优化方案的情况下，优选使用演化过程来推导。

依照本发明的方法可以至少部分地使用例如计算机程序的软件来实现。因而，可以清楚看出，当从本发明的其它方面看时，本发明还提供了计算机软件、计算机程序元件和计算机程序，其中当被安装在数据处理装置上时，所述计算机软件特别适合于执行此处所描述的一个或多个方法；当在数据处理装置上运行所述程序元件时，所述计算机程序元件包括用于执行此处所描述的一个或多个方法的计算机软件代码部分；并且当在数据处理系统上运行所述程序时，所述计算机程序包括适合于执行此处所描述的一个或多个方法的所有步骤的代码装置。本发明还扩展到一种计算机软件载体，所述计算机软件载体包括软件，当使用该软件进行电子线路设计或构造时，包括数据处理装置的系统使得所述系统连同所述数据处理装置一起执行本发明的方法步骤。这种计算机软件载体可以是物理存储介质，诸如 ROM 芯片、CD-ROM 或盘，或者可以是信号，诸如经由导线的电子信号、光信号或来自诸如卫星的无线电信号等。

还将理解的是，并非本发明的方法的所有步骤都需要由计算机软

件来执行，并且由此根据本发明的更宽的方面，提供了计算机软件，并且这种软件安装在计算机软件载体上以便执行此处阐述的方法步骤的至少一个。

因此，本发明可适当地被实施为供计算机系统使用的计算机程序产品。这种实施方式可以包括要么是固定在有形介质上的计算机可读指令序列，所述有形介质诸如是计算机可读介质，例如磁盘、CD-ROM、ROM 或硬盘，要么是在包括但不限于光学或模拟通信线路的有形介质上或无形地使用包括但不限于微波、红外或者其它传输技术的无线技术，经由调制解调器或其它接口设备传输至计算机系统的计算机可读指令序列。所述计算机可读指令序列实施了此处预先描述的所有或部分功能。

本领域技术人员将理解的是，这种计算机可读指令可以依照很多种编程语言来写入，以供许多计算机体系结构或操作系统使用。此外，可以使用当今或未来的任何存储器技术来存储这种指令，包括但不限于半导体、磁性、或光学，或者可以使用当今或未来的任何通信技术来传输这种指令，包括但不限于光学、红外或无线电。预计，这种计算机程序产品可以作为带有印刷或电子文档的可移动介质（例如，收缩薄膜软件）分送，利用计算机系统预先加载在例如系统的 ROM 或固定的盘上，或经由网络（例如，因特网或全球网）从服务器或电子公告板分送。

附图说明

现在将仅以举例形式，并且参考附图来描述本发明的多个优选实施例，其中：

图 1 示意性地示出了数字电子线路的设计过程；

图 2 示意性地示出了示例性的演化算法过程；

图 3 示意性地示出了本发明的实施例；

图 4 和 5 依照本发明实施例示意性地示出了对用于优化电路的优化方案的使用；以及

图 6 示出了本发明实施例的优化性能。

具体实施方式

现在将参考 Berkeley SIS 电子线路设计自动化系统来描述本发明的优选实施例。所述 SIS 系统被描述在例如 E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton 和 A. Sangiovanni-Vincentelli 的 SIS: A system for sequential circuit synthesis (用于时序电路合成的系统), 1992 年在位于 Berkeley 的加州大学发表的 Technical Report UCB/ERL M92/41; 以及 1992 年在位于 Berkeley 的加州大学的电机工程和计算机科学部的博士论文中由 E. M. Santovich 发表的 Sequential Circuit Synthesis at the Gate Level。

(然而应当注意的是, 虽然当前的实现方式是参考 SIS 系统来描述的, 但是正如本领域技术人员将理解的, 本发明不专用于 SIS 系统, 而是也可应用于并且供其它电子设计自动化技术使用。)

正如本领域中所公知的, 在实际实施和文献方面, SIS 软件可以免费获得并且被广泛使用。因此, 它是众所周知的并且非常了解的电子线路设计自动化工具。SIS 可以执行与技术无关的优化、技术映射和与技术有关的优化。

SIS 优化方案可作为“脚本”给出, 所述脚本作为文本文件提供给 SIS 软件。通过理论和人工实验, 大量的努力用于探究为 SIS 系统推导良好的通用脚本, 并且在 SIS 软件分送中包括脚本的选择。SIS 提供了许多不同的优化命令, 那些命令会出现在其优化脚本中, 并且这些命令中有许多采取用于微调其特性的数值参数和选项标志。

最广泛使用的三个 SIS 脚本是 script.rugged、script.algebraic 和 script.boolean。下面阐明这些脚本:

script.rugged:

```
# Initial pre-processing: try both with this and without this command:  
full_simplify
```

```
REPEAT until no further improvement possible, keeping best result:  
{
```

```
sweep
eliminate -1
simplify -m nocomp
eliminate -1

sweep
eliminate 5
simplify -m nocomp
resub-a

fx
resub -a
sweep

eliminate -1
sweep
full_simplify -m nocomp
}
```

script.algebraic:

```
# Initial pre-processing: try both with and without this command:
full_simplify
```

REPEAT until no further improvement possible, keeping best result:

```
{
    sweep
    eliminate 5
    simplify -m nocomp -d
    resub -a

    gkx -abt 30
    resub -a
    sweep
    gcx -bt 30
}
```

```
    resub -a
    sweep

    gkx -abt 10
    resub -a
    sweep
    gcx -bt 10
    resub -a
    sweep

    gkx -ab
    resub -a
    sweep
    gcx -b
    resub -a
    sweep

    eliminate 0
    decomp -g *
}
```

script.boolean:

#Initial pre-processing: try both with and without this command:

full_simplify

REPEAT until no further improvement possible, keeping best result:

```
{
    sweep
    eliminate -1
    simplify
    eliminate -1

    sweep
    eliminate 5
```



```
simplify

resub -a

gkx -abt 30
resub -a
sweep
gkx -bt 30
resub -a
sweep

gkx -abt 10
resub -a
sweep
gkx -bt 10
resub -a
sweep

gkx -ab
resub -a
sweep
gkx -b
resub -a
sweep

eliminate 0
decomp -g *

eliminate -1
sweep
}
```

使用 SIS 脚本的公共优化策略在于试验 `script.rugged`，并且如果这样未能在可利用的时间内产生结果，则使用 `script.algebraic`。

在本实施例中，除了非常微小的增加以外，标准 SIS 1.3 软件在未

改变的状态下被使用，以便允许执行次数被更准确地测量，并且允许记录优化结果以便易于由演化算法软件（其是完全独立的）访问。

图 3 依照本发明示意性地示出了为电子电路设计推导优化方案、然后使用那些方案来优化待设计的电子电路的系统。

如图 3 所示，并且如上所述，本实施例的系统可以被认为是划分为两个不同的部分或阶段，第一是训练阶段 10，其中使用演化算法来推导所选电子电路的优化方案，其次，第二是操作阶段 11，其中使用所推导的优化方案来优化新的待设计的电子电路。

如图 3 所示，用于推导优化方案的训练阶段 10 包括很多步骤。

首先，提供所选的个体电路或感兴趣的较小电路组作为优化方案推导过程的输入（步骤 12）。

然后使用演化算法为所输入的电路和电路组的每一个生成专门的并且相对快速的优化方案（步骤 13）。这样提供一组已演变的快速优化方案，这些优化方案是为所输入的个体电路或电路组专门推导出来的（步骤 14）。

然后选择已演变的专用优化方案的一个或多个以便形成一套优化方案（优化方案集）（步骤 15），该套优化方案于是将用于优化待设计的新电路。

可以如所期望地选择把哪些优化方案包括在一套优化方案 15 中以供使用。例如，这可以基于对除目标电路（即，辅助电路）以外的电路数目的估计，其中给定的优化方案将为所述目标电路提供改进的优化性能，和/或基于该优化方案和为其提供改进优化性能的电路对于该套优化方案中存在的其它优化方案补充到多好的程度。

如图 3 所示，例如可以采用由其它方法生成的优化方案来扩充该套已演变的优化方案 15，所述其它方法例如人工设计方案，包括标准通用方案 16 或演变得更通用的方案 17。

在步骤 13 使用的用于为每一电路（或电路组）演变优化方案的演化算法可以采取任何适当的形式。图 2 示意性地示出了演化（遗传）算法的基本操作。依照这种方式操作的演化算法适合于在本发明中使用。

如图 2 所示，所述演化算法基本上通过采取候选解决方案初始集

（即，此情况下的优化方案）来操作（步骤 21），然后根据期望的任务来评估候选解决方案的性能（通常被称为测量候选解决方案的“适合度”）（步骤 22）。然后，丢弃基于这种评估发现的具有最差性能的候选解决方案（步骤 23），并且选择被发现能够执行的更好的候选作为“母版”以用来演变新的、有希望改进的候选解决方案（步骤 24）。

然后组合所选的“母版”候选解决方案和/或采用某种方式改变（诸如随机地）以便形成某些新的候选解决方案（通常被称为“后代”候选）（步骤 25），然后亲自评估最新演变的候选解决方案并且重复所述过程直到到达某个已定义的结束点为止（步骤 26）。

在本实施例中，所使用的演化算法是没有卓越特征的遗传算法。这种遗传算法例如在如下文献中描述过：J. H. Holland 于 1975 年在 Michigan 大学出版社发表的 *Adaption in Natural Artificial Systems*, Ann Arbor 以及 D. E. Goldberg 于 1989 年在 Addison Wesley 发表在 *Genetic Algorithms in Search* 上的 *Optimisation and Machine Learning*。

线性使用的遗传算法利用尖端钝化和尖子主义把选择分等，所作用于的总数为 30。每一演化从不同的原始优化方案群体开始运算，每一优化方案随机产生并且正好是三个命令长。（当然，如果期望，还可以预先存在的结果（无论是例如手工设计还是预先演变。）为根源而逐渐发展演化运算）。

为了评估优化方案的符合程度并且把它们排序，首先根据由每一优化方案提供的优化结果（即，优化质量度量）来比较各个优化方案。如果各个方案的优化质量等同，那么按照优化方案（脚本）直到终止所花费的时间（促进更快速的方案）把候选方案分等级。如果仍然存在约束，那么给最短的候选方案分配最高的等级。（正如本领域中所公知的，处理具有固定优先级的多个符合程度准则的这种方法常常被称为词典编纂的或字典顺序）。

所述候选优化方案通常经由几百代演变而来，以便提供最终输出的已演变的优化方案。在这个实施例中，在演变候选优化方案的过程（图 2 中的步骤 25）中所允许的“遗传”变量如下：

Mutation-New Command: 方案的命令被替换为新的随机命令，并且还随机地选择任何参数。

Mutation-New Command Variant: 用于定义命令的特殊版本的符号标志随机变化。如果与所述命令相关联的数值参数的数目没有因此改变，并且那些参数的有效范围也没有改变，那么所述数值参数保留不变。否则，随机产生与新的命令变体相关联的所有数值参数。

Mutation-Parameters : 通过繁殖遗传算法（breeder genetic algorithm, BGA）变异方法来调节命令的数值参数之一（参见 1994 年 H. Mühlenbein 和 D. Schierkamp-Voosen 在 Evolutionary Computation 的 1(4):335-360 中发表的 The Science of Breeding and its Application to the Breeder Genetic Algorithm BGA）。

Homologous crossover: 标准两点交叉，始终保持参数不与相关联的命令分离。

Nonhomologous crossover: 如同上述，不过交叉点之间的部分随机移位。

Insert: 在随机位置插入一个新的随机命令，用于随机产生任何参数。

Delete: 删除一个随机选择的命令。

Block insert: 选择在一个母版方案中处于随机位置并且具有随机长度的命令的连续序列，并且将其插入到第二母版中的随机位置（增加方案长度）以便产生后代。

Block delete: 选择处于随机位置并且具有随机长度的命令连续序列并且删除它。

这些“遗传”变量操作符被配置为只使用整数从事优化方案的直接数字表示。对不同遗传变量操作符的概率选择是通过实验推导的。当然，其它技术也是可能的。用于改变优化方案长度的操作符的使用概率被平衡，以致不存在对优化方案的嵌入趋势，用于在没有选择的情况下成长或收缩。所使用的数字表示被选择以便保证每一 SIS 命令将具有有效的参数集，即，使得优化方案在演化期间不可能无效。

使用上述的选择（符合程度）准则，上述遗传算法可用来为所选的感兴趣的电路（目标电路）演变优化方案。在演化运行期间会发现，甚至在优化质量停止提高之后，通常还会存在优化速度被提高的最终阶段。

现在转向图 3 所示的操作阶段 11，也就是使用该套所选的优化方案来优化新电路，如图 3 所示，此过程具有三个步骤。首先，在步骤 18 输入待优化的新电路。然后使用来自该套优化方案 15 的多个优化方案（例如顺次或并行）来试图优化新电路（步骤 19），并且把最佳优化结果作为对所述电路的优化方案（步骤 20）。换言之，该套中多个所选的专用优化方案被试验，直到获得良好的优化结果为止。

在此实施例中，为每一个待优化的新电路试验该套优化方案 15 中的所有优化方案。可选地，例如可以基于对电路的一个或多个特殊类型的优化方案的已知性能来进行对优化方案的某种形式的选择以便试验。

如图 3 所示，为需要试验的每一优化方案设置时间限制，一旦到达时间限制时，就可获取最佳结果。这样有助于确保处理过程的综合效率（不过并不必要）。

在本实施例中，当正在操作阶段使用优化方案来优化新电路时，依照不同于它们在训练阶段执行的方式来执行所述方案。这是考虑到，在训练阶段期间，演化过程致力于开发具有单一目标的优化方案，所述单一目标是用于优化其专用目标电路。然而，在操作阶段中，优化方案将被用来优化不同于它们的目标电路的电路，即，并非为了此目的而设计的电路。因此，在操作阶段依照稍有不同的方式使用优化方案十分有益。

考虑于此，在操作阶段中，宁可在优化方案的每一步骤之后测量优化的质量（例如通过包括用于输出适当的质量度量的命令（诸如下面论述的 `SIS print_stats-f` 命令）），也不会方案执行结束时简单地评估优化方案的结果，并且可获得最佳结果。可以将这种配置称为“单步执行”，因为在方案的每一单个步骤之后评估优化方案的结果，而非在其结束时简单地进行。

例如，可以在操作阶段期间允许优化方案的迭代（即，重复执行），所述迭代每次以至此获得的最佳结果开始。在训练阶段期间不必这样做，因为单个优化方案可以增加长度，以便适应在单个迭代内重复。然而，当在操作阶段期间把为给定目标电路特别演变的优化方案应用于其它电路时，迭代十分有益。

在操作阶段 11 的步骤 19 对每一待优化的新电路试验多个优化方案的方式可以如所期望地选择。图 4 和 5 例示了这种配置的两种可能的候选方式。

在图 4 所示的配置中，每一优化方案 30 并行应用于待优化的新电路 18，每一方案 30 基于待优化电路的相同初始描述开始工作。然后获取最佳结果 20。这种配置例如可以在并行操作的多个处理器上执行，以便加速其执行。

图 5 示出了另一可选的配置，其中优化方案 15 被顺序应用于待优化的新电路 18，每一新的优化方案基于通过任何预先试验的优化方案发现的最佳结果开始工作。因而，选择对电路 18 试验的优化方案（步骤 31），然后应用于至此发现的最佳结果（步骤 32），然后是所选择和使用的新的优化方案等，直到选择了最终结果 20 为止。在此配置中，例如，可以选择试验优化方案的顺序，由此可进一步增强优化处理。

本实施例的电路优化系统的性能可与使用 SIS 系统提供的标准优化方案（脚本）的 SIS 系统的性能相匹敌。对于这种性能评估，从广泛使用的 MCNC'91 基准电路集中（参见 S. Yang 于 1991 年在邮政信箱 12889 的 Microelectronics Center of North Carolina, Research Triangle Park, NC 27709 的技术报告中发表的 Logic synthesis and optimisation benchmarks user guide version 3.0；以及 N. Whitaker 在英国曼彻斯特大学计算机科学部 MINT 小组的技术报告 STEED/TI/01/4 中发表的 Status report on EDA benchmarks）取出七十四个测试电路。这些电路如下面的表 1 所示。

表 1：来自 MCNC'91 基准套件的测试电路集

	名称	类型	输入	输出
0	majority	表决器	5	1
1	bl	逻辑	3	4
2	C17	逻辑	5	2
3	cm82a	逻辑	5	3
4	parity	逻辑	16	1
5	tcon	逻辑	17	16
6	cm151a	逻辑	12	2
7	cmb	逻辑	16	4
8	cm150a	逻辑	21	1
9	mux	多路复用器	21	1

10	cm85a	逻辑	11	3
11	cm163a	逻辑	16	5
12	cm138a	逻辑	6	8
13	x2	逻辑	10	7
14	il	逻辑	25	16
15	pml	逻辑	16	13
16	cu	逻辑	14	11
17	pcl	逻辑	19	9
18	cm42a	逻辑	4	10
19	z4ml	2 位加法器	7	4
20	cm162a	逻辑	14	5
21	unreg	逻辑	36	16
22	cc	逻辑	21	20
23	pcler8	逻辑	27	17
24	cordic	逻辑	23	2
25	decod	译码器	5	16
26	sct	逻辑	19	15
27	c8	逻辑	28	18
28	count	计数器	35	16
29	lal	逻辑	26	19
30	b9	逻辑	41	21
31	cht	逻辑	47	36
32	my_adder	加法器	33	17
33	comp	逻辑	32	3
34	i5	逻辑	133	66
35	example2	逻辑	85	66
36	ttt2	逻辑	24	21
37	i3	逻辑	132	6
38	x1	逻辑	51	35
39	apex7	逻辑	49	37
40	term1	逻辑	34	10
41	frg1	逻辑	28	3
42	i2	逻辑	201	1
43	x4	逻辑	94	71
44	C880	ALU 和控制器	60	26
45	x3	逻辑	135	99
46	apex6	逻辑	135	99
47	rot	逻辑	135	107
48	i6	逻辑	138	67
49	C499	纠错	41	32
50	9symml	计数	9	1
51	frg2	逻辑	143	139
52	vda	逻辑	17	39
53	i7	逻辑	199	67
54	pair	逻辑	173	137
55	C2670	ALU 和控制器	233	140
56	i9	逻辑	88	63
57	C5315	ALU 和选择器	178	123
58	C6288	16 位乘法器	32	32

59	C3540	ALU 和控制器	50	22
60	C7552	ALU 和控制器	207	108
61	C1355	纠错	41	32
62	C1908	纠错	33	25
63	C432	优先权译码器	36	7
64	k2	逻辑	45	45
65	alu2	ALU	10	6
66	alu4	ALU	14	8
67	des	数据加密	256	245
68	dalu	专用 ALU	75	16
69	t481	逻辑	16	1
70	i10	逻辑	257	224
71	i4	逻辑	192	6
72	i8	逻辑	133	81
73	too_large	逻辑	38	3

为了测量标准 SIS 方案的性能，script.rugged、script.algebraic、script.boolean 和 full_simplify 的每一个独自地在每一电路上运行。所使用的计算机是 2.2 GHz 的运行 Linux 的英特尔奔腾 PC，具有 512 兆字节的内存。

有和没有初始 full_simplify 命令，每一脚本都被独立试验。每一组合在每一电路上允许高达 24 小时的处理器时间，或者 36 小时的实际（挂钟）时间，看哪一个是最短的。如果没有对初始电路做出任何改进，则始终采取脚本结果作为该初始电路。所述电路以将该电路包括在 SIS 软件分布程序中的形式输入 SIS。所述“重复直到没有进一步改进可能”的脚本准则是这样的解释的，即：所述脚本在没有给予改进的第一次迭代之后终止。

对于每一电路 c ，从这些通用 SIS 标准脚本看，本实施例的方法要达到的目标被视为最佳结果，并且被标示为 gps_c 。

此处应当注意，虽然某些脚本是随机的，并且假使没有计算机的随机数发生器，当在相同条件下重复时可以产生稍有不同的结果，但是在不考虑这些的情况下，当前评估的结果被发现是足够重复的。

所述方案用于与技术无关的优化。所选定的优化准则将使因子形式的文字数目最小，正如由 SIS 命令 print_stats-f 所报告的。如上所述，此优化准则提供了对用于表示电路的布尔函数的有向无环图中整个逻辑复杂度的测量。在此评估中，并没有考虑测定其它优化准则，诸如从电路输入到输出的延迟（最长路径），不过如果期望，也可以进行。

类似地，对于这些例子，在演化新的专用优化方案（脚本）期间、即在训练阶段 10 期间所使用的质量度量是当优化方案终止时具有因子形式的文字的计数。换言之，把 SIS 命令 `print_stats-f` 的输出（其使得具有因子形式的文字数目将被计数）用作由所讨论的方案提供的优化质量的测量。当在操作阶段中应用于待优化的新电路时，把已演变的优化方案的此图表与如上所述运行的任何通用标准 SIS 脚本生成的相应图表相比较。

对于演化过程来说，已演变的优化方案和演化运算在具有 256 兆字节内存的 1.6GHz 膝上型 PC 上执行。估计优化方案直到终止所用的最大时间量在演化期间依据正被优化的目标电路是什么而所有变化，但是决不超过 600 秒。上述演化算法被使用，并且因而在此时间限制内，还存在在不牺牲质量的情况下选择尽可能快的优化方案。此处应当注意，在下面的例子中，在已演变的优化方案在某些“辅助”（即，非目标）电路上执行更通用的标准脚本的情况下，这样在 1.6GHz 的 PC 上所花费的时间决不超过 550 秒，并且通常是更少的时间。

对其进行评估比较的示例性训练阶段的第一目标电路是电路 C6288（上面表 1 中的电路 58）。此电路被选作例子，这是因为早已注意到，基于二元判定图对于 SIS 系统和候选优化技术都是十分麻烦的。

本实施例的方法演变了以下快速专用优化脚本 A 以便优化电路 C6288：

脚本 A：

```
eliminate -1 10000 -1
eliminate -1 2989 47
fx
eliminate -1 841 -1
fx
eliminate -1 3631 47
fx
eliminate -1 1077 75
sweep
fx
```

```

simplify -m dcsimp
decomp -g
decomp -g
resub *
eliminate -l 9995 0
simplify -m nocomp -f disj_sup
simplify -i 3 -m dcsimp
eliminate -l 9990 -l
simplify -i 3 -m dcsimp
eliminate -l 9366 0.

```

在测试集中的其它 73 个测试电路的十二个上，还发现脚本 A 也执行的很好（即，优化的很好）。这些电路因此放在脚本 A 的“辅助电路”当中。

下面的表 2 示出了这些电路，对此，已演变的脚本 A 执行的比三个标准 SIS 脚本的任何一个都更好。在表 2 中， gps_c 是三个通用标准 SIS 脚本的任何一个中具有因子形式的最小文字数目，而 $t(gps_c)$ 是所述脚本花费的很短的处理器时间。 evo^A_c 和 $t(evo^A_c)$ 给出了脚本 A 的相应性能。还示出了最慢的标准 SIS 脚本所花费的时间。

对于此分析，Script A 如同标准脚本那样在同一 2.2 GHz 基准 PC 上执行，以便允许直接时序比较。

表 2

电路		通用标准			已演变的	
		gps_c	最佳 $t(gps_c)$	最慢 $t(slowest)$	evo^A_c	$t(evo^A_c)$
6	cm151a	26	0.05	0.06	24	0.02
15	pm1	50	0.09	0.11	49	0.04
26	sct	75	0.19	0.20	71	0.08
27	c8	137	0.23	0.27	131	0.11
29	lal	100	0.20	0.24	94	0.09
30	b9	122	0.17	0.26	119	0.12
39	apex7	243	0.52	0.81	233	0.22
40	term1	142	0.77	0.98	141	0.42
42	i2	213	0.52	2.76	212	8.33
44	C880	408	3.13	3.14	399	5.93
55	C2670	712	14.16	14.16	709	146.8
58	C6288	3295	18.26	21.38	3222	18.35
71	i4	204	0.26	86400 timeout	192	0.33

本实施例也被用来为表 1 中的电路 73 演变快速专用优化方案，即 too_large。下面示出的脚本 B 被演变以优化此电路：

脚本 B：

```
collapse
full_simplify -o 1 -m snocomp -1
fx -oz
full_simplify -d -o 0 -m dscimp
xl_partition -n 18 -M 1 -t
full_simplify -o 0 -m dcsimp
eliminate -1 9395 0
full_simplify -d -o 0 -m dcsimp
full_simplify -o 0 -m dcsimp -1
fx -ol
fx -lz
full_simplify -o 0 -m dcsimp -1
decomp -g
eliminate -1 951 3
eliminate -1 789 19
full_simplify -o 1 -m dscimp
fx -1
full_simplify -d -o 0 -m dcsimp
full_simplify -o 1 -m dcsimp
eliminate -1 9365 8
full_simplify -o 0 -m snocomp
eliminate -1 4988 -1
eliminate -1 1934 10
full_simplify -o 0 -m dcsimp
fx -1
full_simplify -o 0 -m snocomp
eliminate -1 9371 7
fx -olz
fx
eliminate -1 9688 8
```

```

full_simplify -d -o 1 -m snocomp
fx -z
eliminate -l 9963 -l
full_simplify -o 0 -m dcsimp
decomp -q
eliminate -l 2032 6
simplify -l 20: 1 -m nocomp
fx -olz
fx -ol
phase -g
eliminate -l 9838 -l

```

此处应当注意，在此脚本 B 中，已经删节了任何冗余的命令。换言之，当优化电路 too_large 时，在不降低脚本性能的情况下，没有被进一步消除的单个命令。

下面的表 3 示出了用于优化其目标电路 73 的脚本 B 的性能及其辅助电路组（根据表 1 中的电路 74，其中它被测试）。

表 3

电路	脚本终止时的改进(%)	如果单步执行的最佳改进(%)
2	11.1	11.1
6	3.8	7.7
7	26.0	38.0
8	-	7.8
9	-	7.8
10	30.4	30.4
13	4.3	4.3
15	-	4.0
16	12.1	12.1
23	-	1.1
26	-	1.3
27	4.4	4.4
30	1.6	1.6
35	5.3	5.3
36	4.1	4.1
37	-	19.5
40	9.2	9.2
41	5.0	10.0
45	-	2.5
50	15.3	18.6
51	-	1.1
71	2.0	5.9
73	30.1	30.1

表 3 示出了每种情况下三个通用标准 SIS 脚本中最佳的改进百分比（这里所述改进为正向的）。表 3 中的右手列示出了用于优化所讨论的电路的脚本 B 的单步执行估价的性能改进（即，在优化方案中每一步骤之后测量优化质量并且获取最佳结果，而非在方案结束时简单地获取结果，如上所述）。

可以看出，对于很多电路来说，脚本 B 提供了改进的优化性能，特别是在操作阶段中的“单步执行”情况下更是如此。

使用一套包括十个已演变的专用优化方案来执行进一步的性能评估。图 6 中示出了此套优化方案的性能分析结果。

图 6 中的星号示出了该套中十个优化方案实际上为其被演变的电路和条件。这些已演变的优化方案包括上述脚本 A 和 B。

图 6 相对于每一电路最佳的通用 SIS 标准脚本示出了此套优化方案跨越表 1 的 74 个采样电路的性能。在图 6 中，为在每一电路上测试的每一脚本绘制了线条（某些基本重叠）。在图 6 中，“1”的相对大小表示所讨论的电路的最佳通用标准 SIS 脚本的性能。因而，例如，已演变的优化方案的相对大小 0.9 意味着已演变的优化方案的性能超过该电路最佳标准脚本 10%（最坏的结果在图 6 顶端的标度消失）。

从图 6 可以看出，即使采用一套只有十个已演变的优化方案，使用该套优化方案可以获得的优化性能也早已超过一半电路以上的最佳通用标准 SIS 脚本，有时如此显著。

图 6 所示的结果是在操作阶段中为所述优化方案使用单步执行评估程序推导的，如上所述。没有使用方案的迭代和顺序应用，尽管它们可能是所期望的并且会产生改进的结果。

从图 6 还可以看出，单独使用该套已演变的优化方案的性能可能次于某些电路。然而，这例如可以通过把标准通用 SIS 脚本添加到该套优化方案来避免（例如对它们的操作给予适当的时间宽限）。这样把该套优化方案的性能罩在相对大小“1”处（如图 6 中通过处于相对大小 = 1 的水平实线所示）。还可以通过把其它或可选的已演变优化方案添加到该套来改善性能。

虽然已经仅参考单个与技术无关的优化准则论述了上述例子，但是对于依照本发明演变的优化方案而言，还能够跨越通常独立考虑的

优化活动。

例如，还可以执行或考虑后续技术映射，而不是孤立地执行与技术无关的优化，然后把优化的质量度量作为总映射区域，而不是例如采用映射之前的网络的因子形式的文字。除了与技术无关的优化之外，该优化方案接着将考虑映射算法的特性和技术库。

示例性的制造工艺通过利用 SIS 软件分送的 CMOS 标准单元库 stdcell2_2.genlib 映射算法来描述。例如，人们可以预定义所述映射将利用 SIS 命令 map-m 0-AF; phase-g 来执行，其被推荐用于考虑技术库中给出的负载限制的最小面积电路。这样是允许的，即，在优化方案演化中通过在演变期间把这些命令添加到每个优化方案并且采取最终映射的区域作为优化质量。

下面示出了在映射到库 stdcell2_2 的情况下、依照这种方式演变的优化方案用于优化电路 my_adder:

```
simplify -m dscimp
map -f 3 -B 0 -1
decomp -q
tech_decomp -a 2 -o 2
xl_partition -t
simplify
simplify -m dcsimp
fx -z

# Predetermined final mapping:
map -m 0 -AF; phase -g
```

在此脚本中，已演变脚本的第二个命令执行对技术库的初级映射。后续命令对技术库中的组分和表示电路的有向无环图中的节点之间的理想对应进行破坏。然而，尽管如此，如果删除初级映射命令，那么将发现预定义的最终映射之后的区域会增加。与最终映射之前其被简单用于优化具有因子形式的文字相比，所述脚本因此好像更加考虑了映射处理。

如果期望，考虑了诸如此方案的额外或稍慢优化准则的优化方案也可以包括在一套优化方案中以供使用，由此潜在地给予了更具有“技术意识”的优化。

一旦电路已经被优化，那么就可以例如使用本领域公知的已知技术来构造。

从上文可以看出，至少在其优选实施例中，本发明提供了用于优化待设计的电子电路的系统，所述系统在已知的现有优化技术之上提供了改进。它提供了改进了的工具和技术以用于设计和构造电子电路。

特别的，这通过演变一套专用优化方案，然后依次使用所述方案来优化待优化的新电路，把最佳结果作为最终的优化来获得。

申请人已经具体认识到，如果人们演变脚本，所述脚本足以快速操作以便允许充分演化，那么通常相对易于生成专用优化方案，这些专用优化方案与任何已知的标准通用脚本相比，为其目标电路执行的更好。此外，较好的已演变优化方案实际上比一般的已知通用标准脚本可能花费更少的处理器时间来完成它们的优化。

此外，当应用于某些其它电路时，被演变以用来优化一个特殊电路的优化方案还经常生成较好的结果，尤其是当初始目标电路十分具有挑战性时更是如此。例如，被演变以优化一个相对小的电路的优化方案可能生成甚至用于某些大电路的较好的结果，反之亦然。

这都意味着能够通过演化来开发包括相对较小数目的优化方案的套件，因而这便于在待优化的新电路上依次试验那些优化方案的每一个，并且例如与已存在的已知的标准通用优化方案相比，实际上将倾向于给出改进的优化性能。

申请人还已经发现：为同一目标电路独立演变的优化方案可以具有不同的辅助电路组，并且每个均同样演变以便优化不同目标电路的两个不同的优化方案可以重新具有非常不同的辅助电路组。这意味着，对于只有少许演变的专用优化方案的辅助电路并集实际上有可能覆盖并且为许多电路提供改进的性能。

此外，修改已演变的优化方案以便例如消除冗余的命令也会改变所述优化方案的辅助电路组。类似地，经由演化运算从部分方式选择的优化方案可能具有不同的辅助电路组，与完全为其专用目标电路磨

练的演化运算的最终结果相比，这些优化方案为辅助电路组提供了改进了的优化性能。

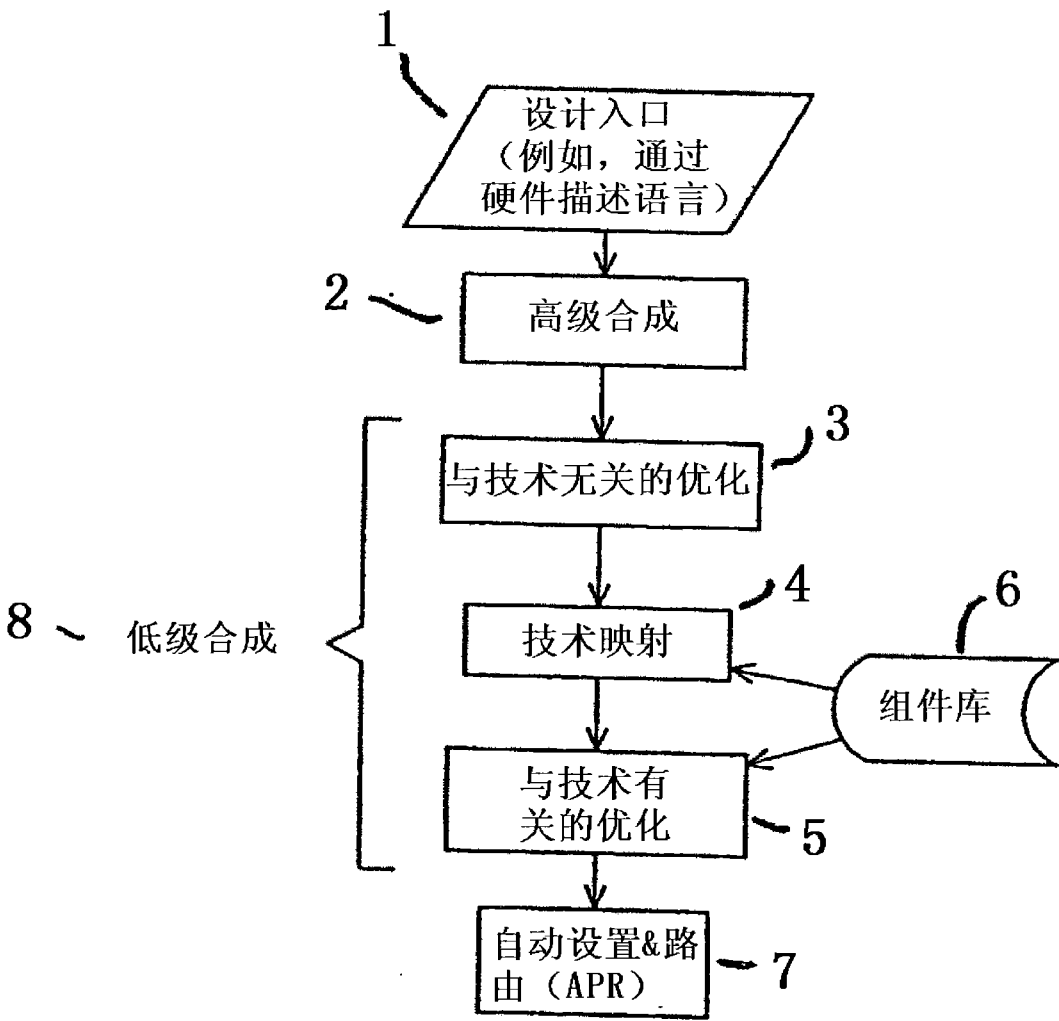


图 1

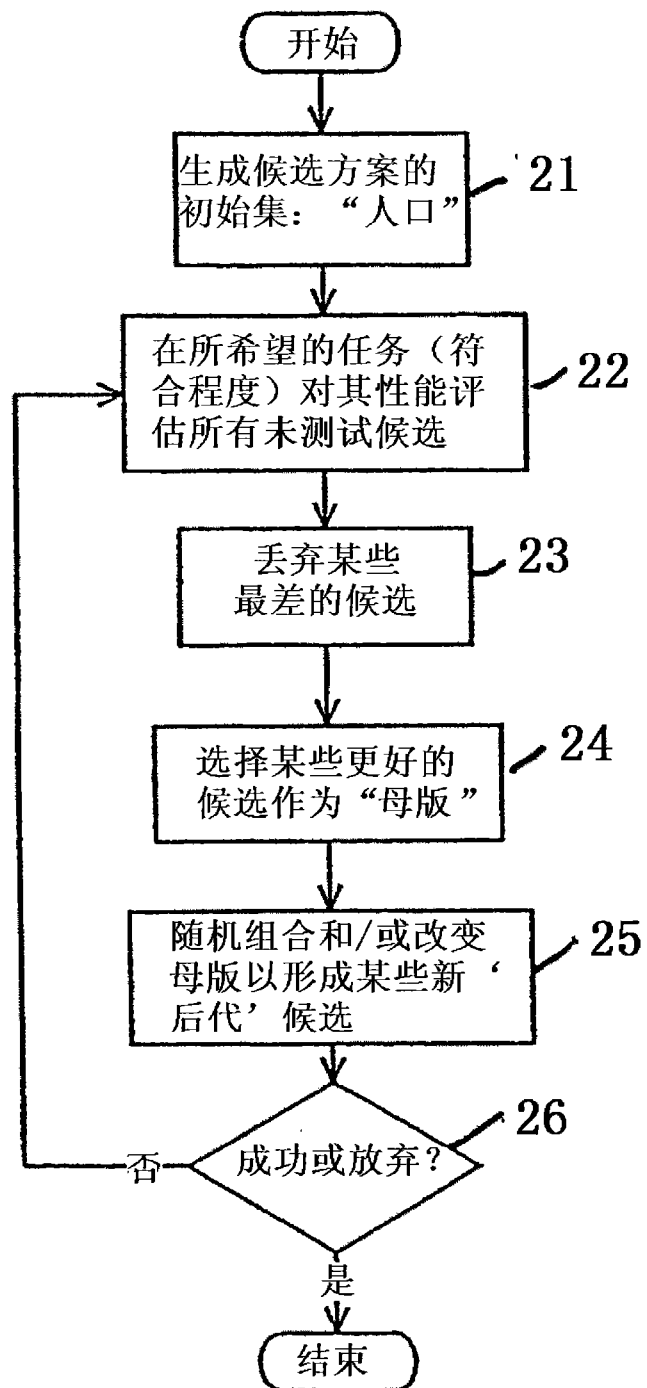


图 2

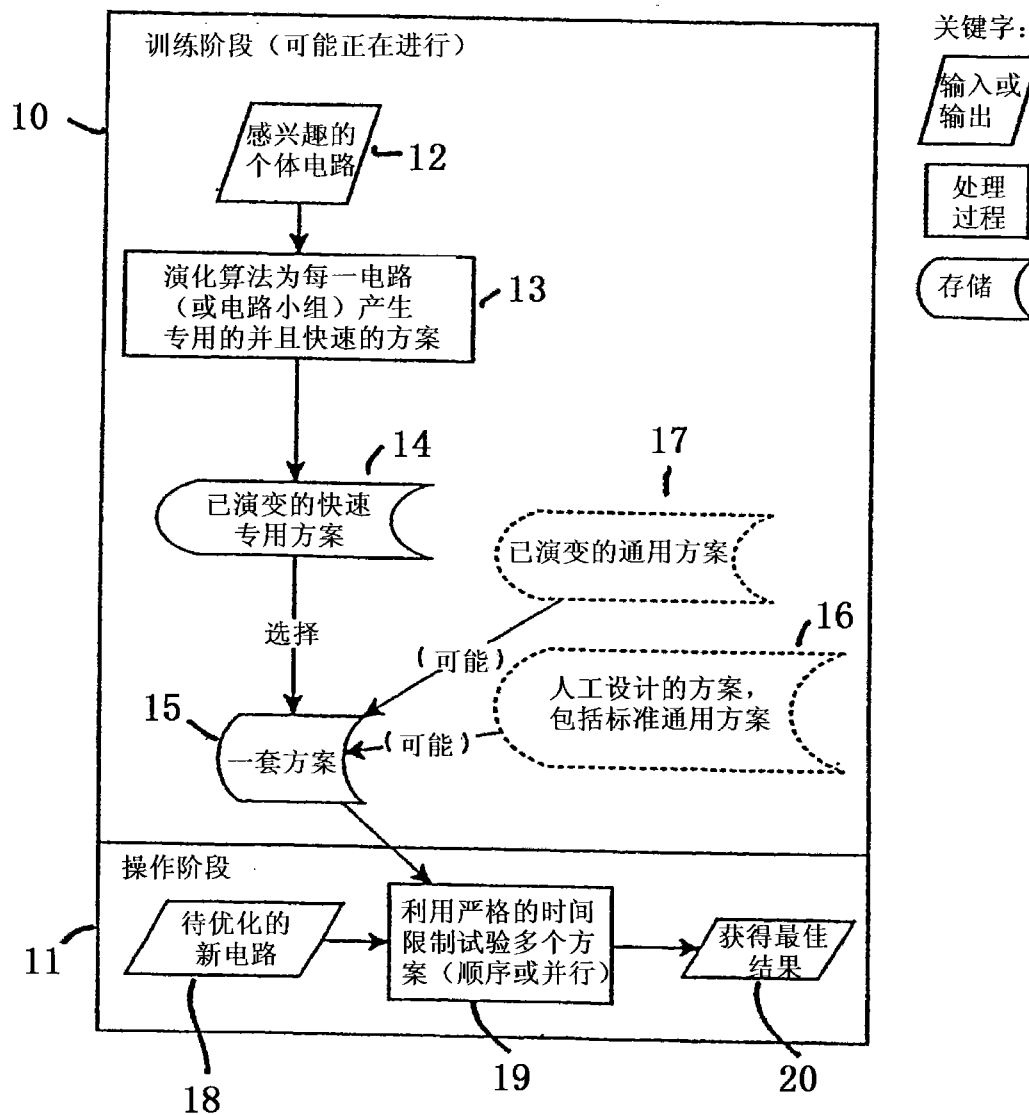


图 3

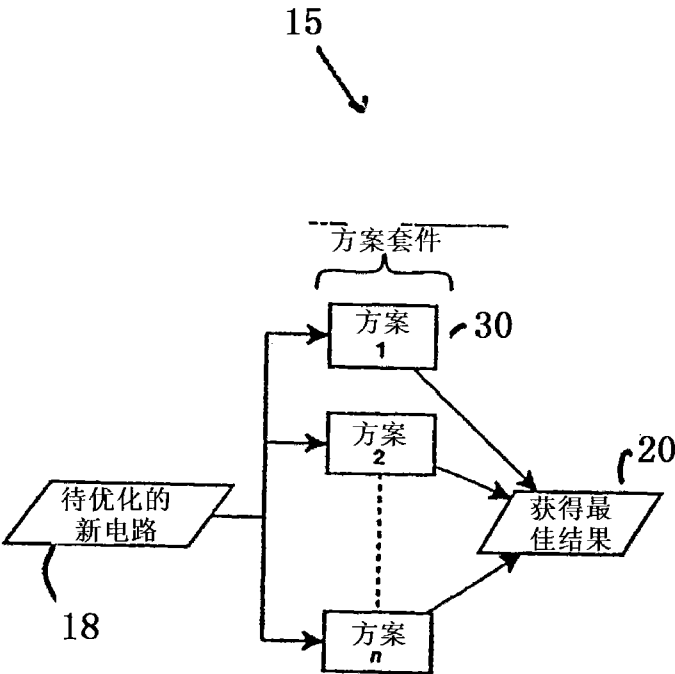


图 4

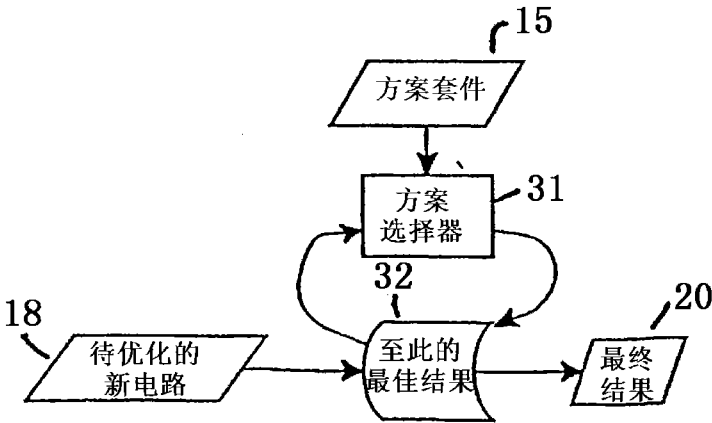


图 5

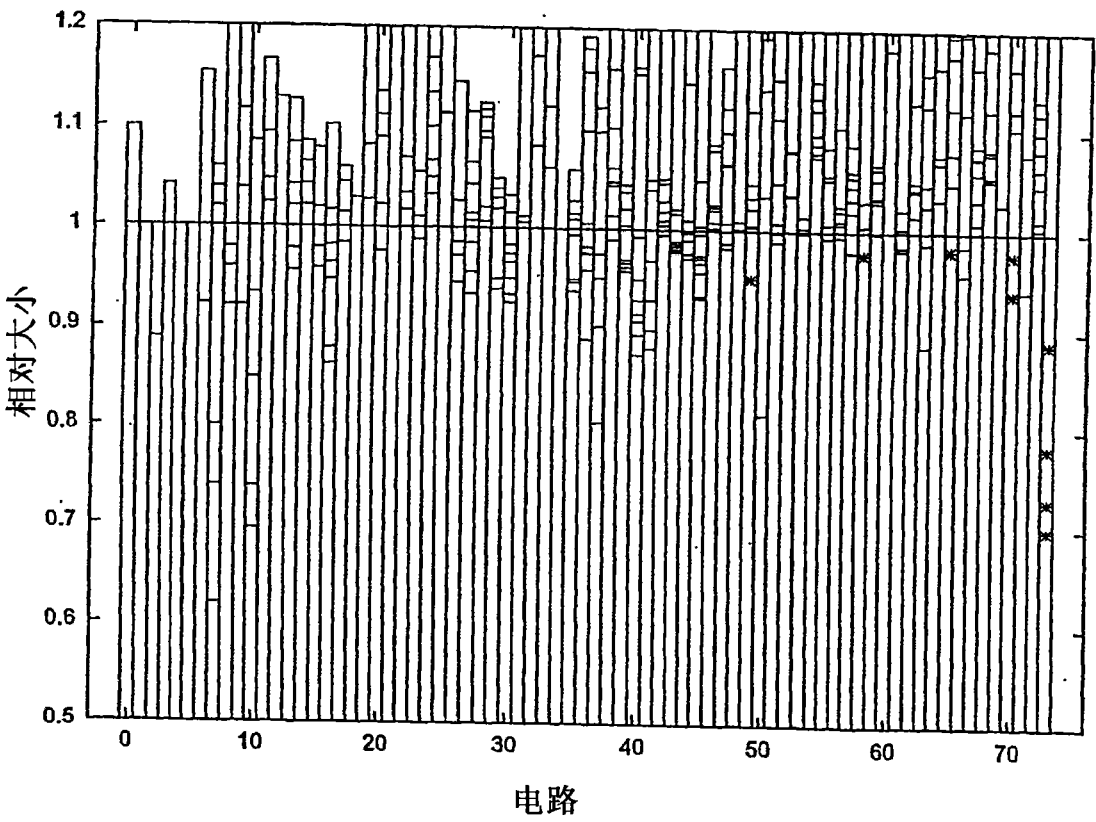


图 6