



(12) 发明专利

(10) 授权公告号 CN 113454471 B

(45) 授权公告日 2024. 10. 01

(21) 申请号 202080015612.1

(22) 申请日 2020.03.13

(65) 同一申请的已公布的文献号
申请公布号 CN 113454471 A

(43) 申请公布日 2021.09.28

(30) 优先权数据
62/817,596 2019.03.13 US

(85) PCT国际申请进入国家阶段日
2021.08.19

(86) PCT国际申请的申请数据
PCT/US2020/022758 2020.03.13

(87) PCT国际申请的公布数据
W02020/186214 EN 2020.09.17

(73) 专利权人 美商新思科技有限公司
地址 美国加利福尼亚州

(72) 发明人 E·吉茨达尔斯基

(74) 专利代理机构 北京市金杜律师事务所
11256
专利代理师 傅远

(51) Int.Cl.
G01R 31/3185 (2006.01)

(56) 对比文件
US 2011191643 A1, 2011.08.04
US 2007220381 A1, 2007.09.20

审查员 魏晓坦

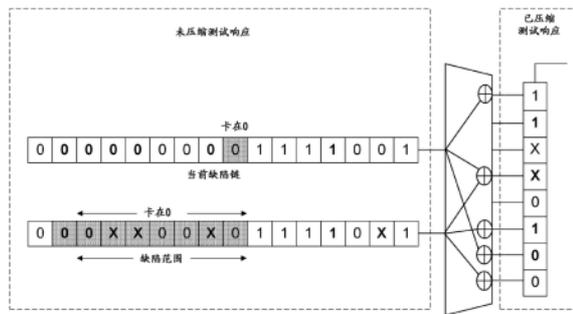
权利要求书2页 说明书19页 附图13页

(54) 发明名称

用于多个链缺陷的单次通过诊断

(57) 摘要

本文中公开了用于电路设计测试中的多个链缺陷的单次通过诊断的方法、系统和存储介质实施例。实施例包括(多个)处理器以在被测电路中选择多个扫描链并且确定至少第一扫描链中的第一缺陷和第一扫描链或第二扫描链中的第二缺陷的存在。多个扫描链可以包括每个具有相应多个扫描单元的特定扫描链。(多个)处理器可以将第一缺陷映射到第一扫描单元的第一范围,并且将第二缺陷映射到第二扫描单元的第二范围。至少部分基于失败捕获模式集合,(多个)处理器可以定位第一范围的第一扫描单元中的第一缺陷以及第一范围或第二范围的第二扫描单元中的第二缺陷。



1. 一种用于被测电路CUT中的多个链缺陷的单个通过诊断的方法,所述方法包括:
由处理器在所述CUT中选择多个扫描链,其中:
所述多个扫描链包括第一扫描链和第二扫描链,
所述第一扫描链包括被连接以形成所述第一扫描链的多个第一扫描单元,以及
所述第二扫描链包括被连接以形成所述第二扫描链的多个第二扫描单元;其中
所述第一扫描链和所述第二扫描链是缺陷扫描链;
由所述处理器确定至少以下各项的存在:
所述第一扫描链中的第一缺陷,以及
所述第二扫描链中的第二缺陷;
由所述处理器将所述第一缺陷映射到第一扫描单元的第一范围,并且将所述第二缺陷映射到第二扫描单元的第二范围;
由所述处理器至少部分基于第一失败捕获模式来定位所述第一范围的第一扫描单元中的所述第一缺陷、以及所述第一范围或所述第二范围的第二扫描单元中的所述第二缺陷;其中
所述第一失败捕获模式基于排名得分被选择,以使得具有最大范围的潜在缺陷扫描链的失败捕获模式被选择;
基于根据所述第一失败捕获模式计算得到的更新后的排名得分,动态选择随后的失败捕获模式;以及
由所述处理器输出与所述第一扫描单元相对应的第一位置和与所述第二扫描单元相对应的第二位置。
2. 根据权利要求1所述的方法,其中所述第二范围比所述第一范围窄。
3. 根据权利要求1或2所述的方法,其中所述失败捕获模式是失败捕获模式集合的元素。
4. 根据权利要求1至2中任一项所述的方法,还包括:由所述处理器标识与所述第一缺陷相对应的第一缺陷类型、以及与所述第二缺陷相对应的第二缺陷类型。
5. 根据权利要求4所述的方法,其中所述第一缺陷类型和所述第二缺陷类型中的每个包括卡在0、卡在1、缓慢上升、缓慢下降、缓慢上升和下降、快速上升、快速下降、快速上升和下降、或其组合。
6. 根据权利要求1至2中任一项所述的方法,其中在至少一个未压缩测试激励中扫描、应用捕获时钟并且扫描出至少一个未压缩有故障捕获测试响应之后,所述失败捕获模式失败。
7. 根据权利要求1至2中任一项所述的方法,其中选择所述失败捕获模式包括:
由所述处理器注入所述第一范围内的第一故障候选;
由所述处理器执行所述CUT的模拟;以及
由所述处理器至少部分基于所述CUT的所述模拟和所述注入,来计算所述第一故障候选的第一排名得分。
8. 根据权利要求7所述的方法,还包括:
由所述处理器至少部分基于无故障CUT的模拟和第二故障候选的注入,来注入所述第一范围内或所述第二范围内的所述第二故障候选,其中所述第二故障候选是通过所述第一

范围或所述第二范围内与所述第二故障候选相对应的多个故障效果的叠加得到的复合故障；

由所述处理器更新所述无故障CUT的所述模拟,以传播基于修改的任何变化;以及

由所述处理器至少部分基于所述无故障CUT的所述模拟和所述第二故障候选的所述注入,来计算所述第二故障候选的第二排名得分。

9. 根据权利要求3所述的方法,其中所述选择、所述确定、所述映射、所述定位和所述输出是在单次通过中对所述失败捕获模式集合执行的。

10. 根据权利要求7所述的方法,还包括:

由所述处理器修改所述第一故障候选的加载值;

由所述处理器更新所述CUT的所述模拟,以传播基于所述修改的任何变化;以及

由所述处理器至少部分基于所述CUT的所述模拟和所述第一故障候选的所述注入,独立于所述CUT的任何测试压缩方案来计算所述第一扫描单元的排名得分。

11. 根据权利要求10所述的方法,其中所述排名得分是被配置为存储至少所述第一缺陷或所述第二缺陷的签名的数据结构的一部分。

12. 根据权利要求7所述的方法,其中所述随后的失败捕获模式是响应于所述排名得分、所述第一范围或所述第二范围的所述映射、更新后的映射、更新后的排名得分、或其组合中的至少一项而动态地选择的。

13. 根据权利要求1至2中任一项所述的方法,还包括:

由所述处理器在模拟所述第一范围之前模拟所述第二范围,其中所述第一范围比所述第二范围窄。

14. 一种在其上存储指令的非暂态计算机可读介质,所述指令在由处理器执行时引起所述处理器实现根据权利要求1至13中任一项所述的方法。

15. 一种系统,包括:

存储器,存储应用;以及

处理器,连接到所述存储器并且被配置为至少实现根据权利要求1至13中任一项所述的方法。

用于多个链缺陷的单个通过诊断

[0001] 相关申请的交叉引用

[0002] 本申请要求于2019年3月13日提交的美国临时专利申请62/817,596的权益,其全部内容通过引用并入本文。

[0003] 版权声明

[0004] 本专利文档的受让人不反对任何人出现在美国专利商标局档案中的专利文档本身或专利申请进行传真复制,但在其他方面保留所有权利受版权保护的包含的作者作品中的任何内容。

[0005] 规范-免责声明

[0006] 在以下“背景技术”、“发明内容”和“具体实施方式”中,标题不应当被解释为必然限制。在以下“背景技术”、“发明内容”和“具体实施方式”中,任何出版物的引用或标识并不表示任何要求保护或描述的实施例作为现有技术的相关性或地位。文本全部为斜体的段落表示多个Synopsys专利规范共有的文本。

背景技术

[0007] 常规的链诊断算法至少包括以下缺点中的任何一个:1) 失败捕获模式最终可能会被多次模拟,因为可以为多个缺陷链选择失败捕获模式并且在多次通过中使用;2) 每个故障候选的外部模拟故障的计算包括计算有故障已压缩测试响应,这是一个耗时的操作,是一种压缩模式;以及3) 在多个链缺陷的情况下,对提高缺陷链的诊断分辨率的失败捕获模式的确定可能很困难。例如,在其中选择缺陷链进行观察的失败捕获模式可能需要提高该扫描链的诊断分辨率。然而,其中可以选择零个或多个缺陷链进行观察的一些失败捕获模式可能不必要地增加诊断分辨率,从而损害具有多个缺陷扫描链的电路中的缺陷的准确性和可观察性。

附图说明

[0008] 以下“具体实施方式”、“附图”、“所附附加图”和“所附权利要求书”表示所要求保护的发明的创新、实施例和/或示例的性质和优点。所有附图仅出于说明的目的表示所要求保护的发明的创新、实施例和/或示例,并不限制所要求保护的发明的范围。这些图不一定按比例绘制并且是本公开的一部分。

[0009] 在图中,相似的组件或特征可以具有相同或相似的标签形式的附图标记(诸如字母数字符号,例如参考数字),并且可以表示相似或等效的功能。此外,可以通过在参考标号后面加上破折号和区分相似组件的第二标号来区分相同类型的各种组件。如果说明书中仅使用第一附图标记,则该描述适用于具有相同第一附图标记的任何一个相似组件,而与第二附图标记无关。附图的简要描述如下。

[0010] 图1描绘了根据一些实施例的具有片上测试数据压缩能力的集成电路的示例框图。

[0011] 图2描绘了根据一些实施例的对于未知位置的有故障测试响应的示例。

- [0012] 图3描绘了根据一些实施例的对于未知位置的有故障测试响应的示例。
- [0013] 图4描绘了根据一些实施例的针对多个链缺陷的故障模拟。
- [0014] 图5描绘了根据一些实施例的错误签名和有故障已压缩测试响应。
- [0015] 图6描绘了根据一些实施例的示例故障模拟过程。
- [0016] 图7示出了如本文中进一步描述的来自描述设计D4的加速的实施例的示例结果。
- [0017] 图8示出了如本文中进一步描述的来自描述设计D5的加速的实施例的示例结果。
- [0018] 图9示出了如本文中进一步描述的来自描述设计D6的加速的实施例的示例结果。
- [0019] 图10描绘了示出根据一些实施例的用于多个链缺陷的单次通过诊断的示例方法的流程图。
- [0020] 图11描绘了根据本公开的实施例的集成电路的设计和制造中的各种操作的流程图。
- [0021] 图12描绘了根据本公开的一些实施例的示例仿真系统的图。
- [0022] 图13描绘了本公开的实施例可以在其中操作的示例计算机系统的图。
- [0023] 在这些不同的图中,为了与公认的工程实践一致,可以省略附图标记;然而,本领域普通技术人员将理解,当在作为整体的图示和描述这些各种图的随附公开内容的上下文中查看时,将容易理解图示的组件。

具体实施方式

[0024] “附图”和以下“具体实施方式”仅通过说明的方式表示创新、实施例和/或示例,在单个实施例中一起描述各种特征、结构或特性以简化本公开。由这样的示例性创新、实施例和/或示例公开的任何元件、过程、机器、系统、制造或组合物的变化将容易被识别并且可以在不脱离所要求保护的内容的原理的情况下用于商业中。“附图”和“具体实施方式”还可以暗示或明确地表示本文中描述的示例性实施例的子集的优点和改进。

[0025] 在“附图”和“具体实施方式”中,可以描述很多具体细节以实现示例性创新、实施例和/或示例中的一个或多个。为了不模糊以下“具体实施方式”中的示例性创新、实施例和/或示例的呈现,本领域中已知的一些处理步骤或操作可以组合在一起用于呈现和说明的目的并且可能不进行描述详细地。然而,本领域技术人员将认识到,这些示例性创新、实施例和/或示例可以在没有这些具体细节或其等同物的情况下用于商业。在其他情况下,没有详细描述公知的过程和设备,以免不必要地混淆这些示例性创新、实施例和/或示例的各方面。在其他情况下,可能根本不描述本领域已知的一些处理步骤或操作。相反,以下描述集中于各种示例性创新、实施例和/或示例的独特特征或要素。此外,虽然该描述可能以单数时态指代结构的一些组件,但在整个附图中可以描绘多于一个组件并且类似组件用类似数字标记。

[0026] 电子设计自动化(EDA)几乎可以应用于半导体行业的任何电路设计项目。在产品被开发之后,EDA工具可以用于定义特定实现,包括用于生产成品芯片的光刻掩模,这一过程可以称为流片。然后将光刻掩模与制造设备一起使用以制造集成电路晶片。测试和诊断可以用于确定缺陷管芯和任何缺陷的位置,这可以称为缺陷定位。

[0027] 批量诊断可以快速提供足够数量的用于系统缺陷的标识的数据。电气故障分析和物理故障分析(PFA)可以用于减少物理面积和收集有关特定缺陷的信息。结果,系统缺陷的

根本原因可以被标识并且用于掩模校正和/或可以增加产率的设计和制造工艺改进的实现。最后,晶片可以被切割、封装和组装以提供用于分配的集成电路芯片。

[0028] 用于使用EDA工具的示例过程开始于要使用集成电路来实现的产品的设计规范。可以应用逻辑设计工具以基于诸如Verilog或VHDL等描述语言来创建高级描述,并且可以在迭代过程中应用功能验证工具以确保高级描述完成设计规范。合成和测试设计工具可以用于将高级描述转换为网表,为目标技术定制网表,并且插入允许测试成品芯片的测试逻辑。

[0029] 典型的设计流程接下来可以包括设计规划阶段,在该阶段,可以构建和分析芯片的整体平面图,以确保可以在高级别实现网表的时序参数。接下来,可以严格检查网表是否符合时序约束和使用VHDL或Verilog在高级别定义的功能定义。在确定网表并且将网表映射到用于最终设计的单元库的迭代过程之后,可以使用物理实现工具进行布局和布线。执行布局的工具在布局上定位电路元件,并且执行布线的工具定义电路元件的互连。

[0030] 然后可以使用提取工具在晶体管级别分析在布局和布线之后定义的组件,并且验证该组件以确保可以实现电路功能,并且可以满足时序约束。可以根据需要以迭代方式重新审视布局和布线过程。接下来,设计可以经过物理验证过程,诸如设计规则检查(DRC)、布局规则检查(LRC)和布局与原理图(LVS)检查,该物理验证过程分析可制造性、电气性能、光刻参数、和电路正确性。

[0031] 在通过设计和验证过程的迭代完成可接受设计之后,诸如上述那些,所得到的设计可以经过分辨率增强技术,该分辨率增强技术可以提供布局的几何操作以提高可制造性。掩模数据可以被准备和流片以用于生产成品。

[0032] 这种使用EDA工具的设计过程可以包括用于测试成品的电路系统,例如,作为用于测试其他电路的测试台电路。集成电路的有效测试可以使用结构化可测试性设计(DFT)技术。特别地,这些技术可以提供对被测电路(CUT)中的所有或一些存储器元件(例如,触发器、锁存器等)的控制和/或观察。示例DFT方法可以基于扫描链。该方法假定在测试期间,存储器元件可以被包括在可以用作移位寄存器的逻辑或物理电子结构中。这种逻辑或物理电子结构可以称为扫描链。因此,所设计的逻辑电路具有两种(或更多种)操作模式,至少包括功能模式和测试模式。在功能模式下,存储器元件可以执行它们的常规功能。在测试模式下,存储器元件可以变成被连接以形成一个或多个扫描链的扫描单元。这些扫描链用于将测试激励扫描输入到CUT中并且扫描输出测试响应。例如,应用测试模式可以包括:执行测试激励的扫描输入(或加载),应用一个或多个捕获时钟,以及执行所捕获的测试响应的扫描输出(卸载)。另外,在一些实施例中,可以将测试响应与无故障测试响应进行比较以确定CUT是否正常工作。

[0033] 一种减少测试引脚、测试数据量和测试应用时间的方法可以是使用已压缩测试激励和已压缩测试响应,而不是分别加载所有测试激励和卸载整个测试响应。图1中示出了具有片上测试数据压缩能力的集成电路的框图。因此,测试器可以经由移位寄存器或移位寄存器功能耦合到包括CUT、解压缩器和压缩器的集成电路。

[0034] 仍然参考图1,例如,用于产率学习的批量诊断可以减少在批量生产之前满足积极的产率提升计划所需要的实际时间。给定测试模式集合的测试器故障以及设计信息,诊断工具可以标识故障候选,该故障候选可以用于隔离这些故障的对应(多个)位置。单个设备

的诊断标注在被映射到多个缺陷位置和缺陷机制时可能因此具有一些模糊性,因为多个缺陷可能相互掩蔽或以其他方式关于单个设备不确定。这种模糊性可以通过可以确定一组常见物理缺陷特征或根本原因的批量诊断、统计分析和PFA来减轻或消除。通过解决这些根本原因的问题,可以提高产率。片上测试数据压缩增加了过程的复杂性,因为CUT中的扫描链可能无法被直接观察到。如本领域普通技术人员将理解的,诊断使用用于逻辑、扫描链、时钟树等中的缺陷定位的专门过程。

[0035] 当可能存在多个链缺陷时,可以在设计制造和产率提升的早期阶段执行批量链诊断。链诊断的输入包括缺陷管芯的设计网表、测试模式集合和测试器故障。链诊断可以使用因果分析以基于故障模型来计算故障候选的故障效果,其中故障模型可以包括至少以下缺陷类型:卡在0、卡在1、缓慢上升、缓慢下降、缓慢上升和下降、快速上升、快速下降和快速上升和下降。测试模式集合可以包括链测试模式和捕获模式。如果测试器上的链测试模式失败,则移位操作可能会因扫描链、时钟树、设置重置树或扫描启用树中的缺陷而受到影响。缺陷链和缺陷类型可以基于链测试模式来标识而无需模拟,如K. Stanley在“High-accuracy Flush-and-Scan Software Diagnostic”(2001年IEEE计算机设计测试第18卷第6号第56-62页)中讨论的。例如,链测试模式扫描输入测试序列“00110011”和扫描输出测试序列“00000000”可以指示在对应扫描链中存在卡在0故障。接下来,可疑扫描单元的初始范围可以包括缺陷链中的一些或全部扫描单元。可以至少部分基于失败捕获模式来执行缺陷定位——这里,捕获模式扫描输入测试激励应用一个或多个捕获时钟和扫描输出捕获测试响应。因果链诊断可以使用故障模拟,该故障模拟可以注入在该范围的第一扫描单元(接近扫描输出)中的故障候选并且计算排名得分。故障模拟可以递增地计算该范围内的其他扫描单元的排名得分。例如,可以基于排名得分来修剪故障候选,并且对于每个模拟通过可以消除具有较低得分的故障候选。在一些实施例中,该过程可以继续直到可以使用所有失败捕获模式或者可以将可疑扫描单元的范围减少到单个扫描单元。

[0036] 对于单个链缺陷,故障模拟可以基于以下步骤计算该范围内的第一扫描单元的故障效果:为所选择的失败捕获模式设置加载值,修改该扫描单元和一些或全部下游扫描单元的加载值,通过良好的机器模拟(例如,已知的正确测试、无故障模拟等)计算损坏捕获值,并且通过修改该扫描单元和被示出为卡在0故障的一些或全部上游扫描单元的卸载值来计算有故障测试响应,如图2所示。注意,对于缺陷位置和一些或全部下游扫描单元,损坏加载值可以为0,对于缺陷位置和一些或全部上游扫描单元,损坏卸载值可以为0。模拟失败可以通过比较预期无故障测试响应和有故障测试响应来计算。排名得分可以通过比较模拟失败与测试器失败来计算。排名得分可以包括至少三个分量中的任何一个:例如,预测的、错误预测的和未预测的测试器故障的数目。理想候选故障可以解释所有测试器故障,不包括任何错误预测的测试器故障或任何未预测的测试器故障。

[0037] 多个链缺陷的链诊断可能需要对一系列可疑扫描单元进行故障模拟。对于每个缺陷链,链诊断可以选择一组失败捕获模式,并且基于其他缺陷链的可疑扫描单元的范围计算每个故障候选的排名得分。首先,可以修改可疑扫描单元及其下游扫描单元的加载值。可以执行良好的机器模拟以基于修改后的加载值计算损坏捕获值。可以通过修改可疑扫描单元及其上游扫描单元的损坏卸载值来计算有故障测试响应。可以基于相应故障的故障效果的叠加来计算多个故障的故障效果。当多个故障的故障效果具有仅值0、仅值1或不同相应

值时,合成的复合故障具有0、1或X。

[0038] 图3描绘了在假定每个缺陷链有单个故障的情况下对一系列可疑扫描单元的有故障测试响应的示例计算。有故障测试响应可以基于该范围内存在一个故障候选的假定来计算。关于(w.r.t.)无故障值的变化可能用粗体标记。缺陷定位可以通过对当前缺陷链的故障候选进行排名来执行,其中每个故障候选的排名得分可以通过比较模拟故障与测试器故障来计算,如在R.Guo和S.Venkataraman的“A Technique for Fault Diagnosis of Defects in Scan Chains”(2001年国际测试会议第268-277页)中讨论的和在Y.Huang、W.-T.Cheng、R.Guo的“Diagnose Multiple Stuck-at Scan Chain Faults”(2008年IEEE欧洲测试研讨会第105-110页)中讨论的。

[0039] 在压缩模式下,可以通过比较压缩器电路的输出处的预期无故障已压缩测试响应和有故障已压缩测试响应来计算外部模拟故障。图4描绘了多个链缺陷的故障模拟。对于给定缺陷链,当一系列可疑扫描单元可以用于其他缺陷扫描链时,可以通过计算每个故障候选的排名得分来执行缺陷定位。有故障未压缩测试响应在图4的左侧示出。有故障压已缩测试响应在图4的右侧示出,与无故障值的变化以粗体标记。W.-T.Cheng、K.-H.Tsai、Y.Huang、N.Tamarapalli和J.Rajski的“Compressor Independent Direct Diagnosis”(2004年IEEE亚洲测试符号第204-209页)修改了被测电路使得压缩器被包括在模拟模型中。这种方法可以适用于计算压缩模式下链诊断的排名得分,如Y.Huang、W.-T.Cheng和J.Rajski的“Compressed Pattern Diagnosis for Scan Chain Failures”(2005年IEEE国际测试会议第751-759页)中进一步讨论的。

[0040] 此外,在Y.Huang、X.Fan、H.Tang、M.Sharma、W.-T.Cheng、B.Benware、S.M.Reddy的“Distributed Dynamic Partitioning Based Diagnosis of Scan Chain”(2013年IEEE VLSI测试研讨会第105-110页)中,提出了一种动态分区以提高批量诊断的产量。平均速度可以提高1.2-13.2倍,而准确度($\leq 4\%$)或分辨率($\leq 0.7\%$)只会出现轻微下降。在S.Kundu、K.Kumar、R.Kumar、R.Kapur的“Diagnosing multiple faulty chains with low pin convolution compressor using compressed production test set”(2017年IEEE国际测试大会)中,提出了一种多通过链诊断算法,以提高多个链缺陷的分辨率。结果,缺陷链的可疑扫描单元的范围可以减小,这又可以减小下一缺陷链的有故障已压缩测试响应中的X(一次通过中的多个缺陷的确定性表示)。下面的清单1中示出了常规链诊断算法:

[0041] 清单1:链诊断

[0042] 1:通过=0;

[0043] 2:确定所有缺陷链的缺陷类型和初始范围;

[0044] 3:while(通过 \leq 1)do{

[0045] 4:for(每个缺陷链)do{

[0046] 5:如果分辨率良好,则继续;

[0047] 6:为该链选择失败捕获模式;

[0048] 7:for(每次模拟通过)do{

[0049] 8:基于其他缺陷链的范围修改加载值;

[0050] 9:执行良好的机器模拟;

[0051] 10:基于其他缺陷链的范围修改卸载值;

```
[0052] 11:for (该范围内的每个故障候选) do {
[0053] 12:修改当前故障候选的加载值;
[0054] 13:传播模拟变化;
[0055] 14:修改当前故障候选的卸载值;
[0056] 15:计算有故障已压缩测试响应;
[0057] 16:比较有故障和无故障已压缩测试响应;
[0058] 17:计算该故障候选的得分;
[0059] 18:}
[0060] 19:更新当前缺陷链的范围;
[0061] 20:}
[0062] 21:}
[0063] 22:通过 = 通过+1
[0064] 23:}
[0065] 单次通过链诊断算法
```

[0066] 描述了一种可以解决常规链诊断的缺点的单次通过链诊断算法。本文中公开的增强技术可以增加批量诊断的吞吐量,同时保持诊断标注的准确性和分辨率。可以至少部分基于选择来模拟失败捕获模式至少一次,该选择可以在模拟期间响应于刺激而被动态地执行。该选择可以对每个模拟通过的失败捕获模式进行排名,使得在某个(一些)模拟通过之后可以减少缺陷链的可疑扫描单元的范围的总和。在一些实施例中,所选择的范围可以限于给定的一组确定的缺陷类型。

[0067] 作为模拟的结果,每个模拟通过的排名靠前的捕获模式可以允许观察缺陷链,增加模拟范围,例如,在扫描链和扫描单元方面,同时减少其他缺陷链的总体发生率。这种单次通过链诊断算法还可以包括用于基于有故障未压缩测试响应的增量变化来计算由于故障而损坏的电路测试值的签名(错误签名)的步骤,该增量变化可以包括测试器故障和模拟故障的数据,从而避免计算已压缩测试响应。下面的清单2中示出了这种链诊断算法的示例。

```
[0068] 清单2:增强链诊断(用于多个链缺陷的单次通过)
[0069] 1:确定缺陷链的初始范围;
[0070] 2:for (每次模拟通过) do {
[0071] 3:选择该范围内的缺陷链的失败捕获模式;
[0072] 4:执行良好的机器模拟;
[0073] 5:for (每个缺陷链) do {
[0074] 6:修改其他缺陷链的加载值;
[0075] 7:传播模拟变化;
[0076] 8:计算其他缺陷链的损坏卸载值;
[0077] 9:for (该范围内的每个故障候选) do {
[0078] 10:修改当前故障候选的加载值;
[0079] 11:传播模拟变化;
[0080] 12:计算当前故障候选的损坏卸载值;
```

- [0081] 13:计算错误签名;
- [0082] 14:基于错误签名计算排名得分;
- [0083] 15:}
- [0084] 16:更新当前缺陷链的范围;
- [0085] 17:}
- [0086] 18:}

[0087] 例如,有故障未压缩测试响应的增量变化 (0→1和1→0) 和 (0→X, 1→X, X→0X→1) 可以分别称为硬模拟失败和软模拟失败。如上定义的错误签名可以包括每个已压缩测试响应的<F/H/S>向量,其中F= {0,1}、H= {0,1} 和S= {0,1} 指示测试器故障、硬模拟故障和软模拟故障的不存在和存在。

[0088] 表1描述了如何基于错误签名间接确定外部模拟故障。错误签名可以为排名得分提供计算效率更高的增量计算。在一些实施例中,错误签名可以基于有故障未压缩静止响应的增量变化,这可以避免计算有故障已压缩测试响应的需要。

[0089] 表1:计算极值故障

[0090]	无故障已压缩值	<F, H, S>		有故障已压缩值	模拟故障
	X	-	-	-	否
0	H 偶数	S=0	0	通过	
[0091]		H 偶数	S≠0	X	0→X
		H 奇数	S=0	1	0→1
		H 奇数	S≠0	X	0→X
	1	H 偶数	S=0	1	通过
		H 偶数	S≠0	X	1→X
		H 奇数	S=0	0	1→0
		H 奇数	S≠0	X	1→X

[0092] 图5示出了给定示例故障候选的错误签名和有故障已压缩测试响应的示例。例如,根据一些示例实施例,有故障压已缩测试响应可以具有标记为从0到13的14位。在该示例实施例中,测试响应的每一位可以由值 {0,1,X, ~ 0, ~ 1} 之一表示。其中X表示未知无故障值(例如,由于多个链缺陷);0表示无故障值和有故障值都为0的情况;1是指无故障值和有故障值都为1的情况;~ 0表示无故障值和有故障值分别为0和1;~ 1表示无故障值和有故障值分别为1和0。

[0093] 根据一些示例实施例,错误签名可以包括五个向量:F、H、S、H*和S*。这些向量可以分别对应于测试器故障、硬模拟故障、软模拟故障、增量硬模拟故障和增量软模拟故障。该示例故障候选的排名得分计算可以包括以下步骤:向量H*和S*可以基于已压缩测试响应的

增量变化来计算;排名得分可以基于向量 H^* 和 S^* 进行更新;向量 H 和 S 可以分别通过向量 H 和 S 与向量 H^* 和 S^* 的叠加来计算;向量 H^* 和 S^* 可以针对下一故障候选被初始化。例如,根据一些实施例,这些计算可以表示为 $H(\text{next})=H \oplus H^*$, $S(\text{next})=S \oplus S^*$ (累积增量变化) $H^*=0$ 和 $S^*=0$ (初始化增量变化)。

[0094] 在本示例中,第8位和第10位可以示出两个特殊条件。对于位8,向量 F 、 H 、 S 、 H^* 和 S^* 分别具有值1、0、1、1和0。例如,如果忽略软模拟故障,则该条件可以指示故障候选可以解释对应测试器故障。这种情况可被视为潜在预测测试器故障。对于位10,向量 F 、 H 、 S 、 H^* 和 S^* 分别具有值1、0、1、1和1。该条件可以表明当前故障候选可以解释对应测试器故障(例如,其中 $F=1$ 和 $H(\text{next})=H \oplus H^*=1$)并且其中软模拟故障数为0(例如, $S(\text{next})=S \oplus S^*=0$)。

这种情况可以被视为预测测试器故障,因为当前故障候选解释对应测试器故障(例如, $F=H(\text{next})=1$)并且软故障数为0(例如, $S(\text{next})=0$)。以这种方式,错误签名可以包括用于有效和准确计算排名得分的完整信息。结果,可以避免计算已压缩测试响应。

[0095] 图6示出了故障模拟过程。一些新的区别可以包括以下内容:可以对所有缺陷链执行一次良好的机器模拟,而常规方法对每个缺陷链执行良好的机器模拟;并且排名得分计算可以基于错误签名,而用于计算排名得分的常规方法可以取决于有故障已压缩测试响应。所提出的算法的其他特征可以包括以下内容:计算无故障未压缩测试响应;基于未压缩测试响应的增量变化计算错误签名;基于错误签名计算排名得分;并且在一些实施例中动态地选择失败捕获模式,从而可以提高所有缺陷的整体分辨率。结果,可以避免计算有故障已压缩测试响应。本领域普通技术人员可以意识到和理解附加特征和优点。

[0096] 实验验证

[0097] 可以基于具有一个或多个编解码器(压缩器-解压缩器)的六种设计的故障注入和测试器故障来执行所提出的链诊断算法的实验验证,使得每个编解码器包括非线性解压缩器和有限线性时间压缩器。本实验中所有六种测试设计的特性如表2所示。可以生成特殊模式以诊断多个链缺陷,从而可以直接针对一个链测试模式和十个捕获模式观察每个链。故障注入的实验结果如表3所示。每个实验可以注入多达10个卡住故障。基线可以是在S.Kundu、K.Kumar、R.Kumar、R.Kapur的“Diagnosing multiple faulty chains with low pin convolution compressor using compressed production test set”(2017年IEEE国际测试大会)中描述的多次通过算法。

[0098] 所有实验结果都可以基于单个线程。如果报告缺陷链中的正确位置,则诊断标注可以更准确。分辨率可以基于所报告的故障候选数来衡量PFA的诊断调用的质量。在一些实施例中,如果可以报告缺陷链的至多三个连续扫描单元(位置),则诊断标注可以被认为是能够进行PFA的(针对PFA是良好的),以说明一个非限制性示例。

[0099] 在一些实验中,如上所述的算法已经达到高达100%的准确度,同时与对应基线多次通过算法相比,分辨率提高了2.16%。这样的实验可以证明,所提出的单次通过算法始终如一地提高了诊断标注的质量。仍然参考表3,最右边的列示出了单次通过算法相对于基线多次通过算法的加速比。实验结果表明,在这些实验中,单次通过算法可以比基线多次通过算法平均快20.22倍。这种加速可以反映使用64位字长而不是32位字长的每次模拟通过的模拟模式数的增加,这可以避免对失败捕获模式进行多次模拟的需要,从而基于本文中的

上述错误签名来更有效地计算排名得分。

[0100] 表2:设计特性

[0101]

设计	#翻转	#链	#编解码器	#扫描输入	#扫描输出	#移位
D1	115K	500	1	10	10	231
D2	165K	500	1	4	1	549
D3	165K	500	1	4	2	275
D4	1.81M	6232	11	52	22	291
D5	3.49M	11547	19	65	26	335
D6	4.71M	16176	13	62	27	291

[0102] 表3:基于故障注入的实验结果

[0103]

设计	#注入故障	#实验	#准确		#能够进行PFA		加速
			基线	增强	基线	增强	
D1	1	100	100	100	94	94	5.02

	2	100	200	200	192	192	6.43	
	3	100	300	300	284	285	6.97	
	4	100	400	400	357	366	7.70	
	5	100	500	500	450	461	8.21	
	6	100	600	600	549	562	9.31	
	7	100	700	700	639	647	12.94	
	8	100	800	800	744	755	9.73	
	9	100	900	900	825	832	11.53	
	10	100	1000	1000	903	915	12.23	
	[0104]	D2	1	50	50	50	33	36
2			50	99	100	71	73	22.99
5			50	247	250	136	151	37.35
10			50	494	500	259	264	59.38
	D3	1	50	50	50	34	36	12.38
		2	50	100	100	76	76	20.81
		5	50	248	250	140	158	40.14
		10	50	498	500	292	306	64.39
	<u>总计</u>	<u>5</u>	<u>1400</u>	<u>7286</u>	<u>7300</u>	<u>6078</u>	<u>6209</u>	<u>20.22</u> (平均)

[0105] 接下来,对实际设计进行了基于测试器故障的验证。该示例实验基于具有多个缺陷链的82个缺陷管芯,单个管芯中有多达32个缺陷链。总之,诊断标注相对于基线没有退化。相比之下,作为一项改进,由于较高得分和/或较少报告故障候选,至少有四个管芯变为能够进行PFA。

[0106] 图7-9分别示出了设计D4、D5和D6的每个管芯的加速。D4、D5和D6的平均加速分别为22X、40X和63X。这些实验可以用于确认所提出的算法可以增加批量诊断的吞吐量,并且可以持续改进通过基线(多次通过)算法得到的诊断结果的质量。

[0107] 图10示出了根据一些实施例的用于多个链缺陷的单次通过诊断的示例方法1000。方法1000可以由处理逻辑执行,处理逻辑可以包括硬件(例如,电路系统、专用逻辑、可编程逻辑、微代码等)、软件(例如,在处理设备上执行的指令)或其组合。

[0108] 方法1000将参考图2-10进行描述。然而,方法1000不仅限于这些示例实施例。方法

1000的步骤可以由耦合到至少一个存储器设备的至少一个计算机处理器来执行。下面分别关于图13的1002和1004来描述示例处理器和(多个)存储器设备。在一些实施例中,方法1000可以使用图13的计算机系统1300来执行,计算机系统1300还可以包括至少一个处理器和存储器,诸如图13的那些。

[0109] 在1002中,至少一个处理器1302可以在被测电路(CUT)中选择多个扫描链。在一些实施例中,多个扫描链可以包括第一扫描链和第二扫描链。例如,第一扫描链可以包括被连接以形成第一扫描链的多个第一扫描单元。同样地,第二扫描链包括被连接以形成第二扫描链的多个第二扫描单元。扫描单元、扫描链及其选择的其他示例在本文中的其他地方进一步详细描述或参考。

[0110] 在1004中,处理器1302可以在CUT中确定第一扫描链中的至少第一缺陷的存在和第一扫描链或第二扫描链中的至少第二缺陷的存在。换言之,CUT中可以存在多个缺陷,无论是在同一扫描链中还是在多个扫描链中。就故障可以发生的程度而言,例如,测试模式、链模式、捕获模式等或其任何组可能由于多个缺陷中的任何一个而失败,由第一缺陷导致的第一故障可能影响同一扫描链或另一扫描链的稍后测试部分,因此如果仅使用常规技术,后续验证可能很困难或不可能。

[0111] 因此,为了1004的目的,根据其中第二故障可以是可能的一些实施例,但是在实际测试结果可能由于第一故障而被掩盖的情况下,这个未知值的存在也可以被视为故障,诸如第二故障。在一些实施例中,虽然整体模拟状态、电路输出或引出线、位向量或其元素(一些非限制性示例)可能仍然由于多个故障的叠加而不确定,但是在良好的机器模拟的类似部分中,可以通过与已知无故障状态的可观察偏差来检测多个故障。在本文中描述的任何这样的情况下,可以以至少这些方式中的任何一种来确定多个故障。故障检测和缺陷确定的其他示例在本文中的其他地方进一步详细描述或参考。

[0112] 在1006中,处理器1302可以将第一缺陷映射到第一扫描单元的第一范围,并且可以将第二缺陷映射到第二扫描单元的第二范围。这样做时,处理器1302可以有效地缩小CUT的范围以在单次通过中搜索解决多个链缺陷。例如,通过将扫描单元的范围映射到CUT的相应区域(例如,相对于用于测试的设计)到可能存在缺陷的潜在区域或位置,处理器1302也可以由此排除考虑不太可能或不可能导致故障或存在缺陷的CUT的其他部分。根据一些实施例,基于向量的技术可以用于该映射。将扫描单元的范围映射到缺陷或潜在缺陷的其他示例在本文中的其他地方进一步详细描述或参考。

[0113] 选择的目的是尽快缩小范围以减少未知值。失败捕获模式可以在多个模拟通过中进行模拟(例如,每个模拟通过有20个模式)。因此,动态选择以及失败捕获模式的顺序可能会对结果产生显著影响。动态选择可能涉及在模拟的每个步骤中选择具有最大范围的潜在缺陷扫描链的失败捕获模式。关于失败捕获模式的顺序,如果在可比较的模拟中要以相反顺序从一个模拟通过到另一模拟通过模拟模式,则可能需要多次通过,以便为其他等效模拟产生相同结果。这里的“单次通过”可以表示每个失败捕获模式可以被模拟一次。然而,在一些实施例中,每个失败捕获模式的“单次通过”仍然可以包括多个模拟通过,因为并非所有失败捕获模式都可以在单个模拟通过中被模拟。

[0114] 例如,在一个非限制性示例中,如果失败捕获模式的数目是300,但在每个模拟通过中只能模拟多达20个捕获模式,则该算法将需要至少15个模拟通过来模拟所有失败捕获

模式。在另一示例中,如果单独的第一模拟通过将第一范围减少到2-8并且将第二范围减少到3-99,则单独的第二模拟通过可以将第一范围减少到2-8并且将第二范围减少到15-58。如果在第一模拟通过之前应用第二模拟通过,则结果可以得到改善,因为第二(更宽)范围分别包括在第二模拟通过和第一模拟通过之后的44个和97个扫描单元。

[0115] 因此,预期在第一模拟通过之后的X可以比第二模拟通过中的少,因为与第一模拟通过相比,在第二模拟通过之后的第二范围可以较窄。结果,与第一模拟通过相比,第二范围的复合故障在第二模拟通过之后包括更少故障。在一些实施例中,可以在每次模拟通过中动态地选择失败捕获模式,使得在有可能减少较窄范围的失败捕获模式之前选择有可能减少最宽范围的失败捕获模式以进行模拟。如果在每次模拟通过中在较窄范围之前模拟较宽范围,则可以进一步提高诊断分辨率。范围可以基于例如在每次模拟通过之后计算的排名得分来确定,如本文中的其他地方关于动态选择所描述的。在一些实施例中,可以基于缺陷到扫描链中的扫描单元范围的初始映射或更新后的映射来计算排名得分,这又可以影响随后的动态选择。所描述的动态选择的效率可以在每次模拟通过之后基于所有范围内的故障候选的总数来评估,举一个非限制性示例。这里,失败捕获模式的动态选择被认为是动态的,因为它可以反映每次模拟通过之后扫描链的更新后的范围。

[0116] 在1008中,处理器1302可以至少部分基于失败捕获模式集合来定位第一范围的第一扫描单元中的第一缺陷。类似地,根据一些实施例,处理器1302还可以至少部分基于失败捕获模式集合来定位可以在第一范围或第二范围中的第二扫描单元中的第二缺陷。因此,以这种方式定位缺陷可以允许将范围(例如,1006中的每个映射)定制为适合于更有效地定位和解决更多复合缺陷的可管理尺寸。同样,CUT的可能几乎没有缺陷的部分可以对其避免使用宝贵的测试资源(1006中的每个映射),从而在1008中在需要以更高精度和效率定位缺陷时提高分辨率。模仿特定ATE行为的基于事件的测试模式的其他示例在本文中的其他地方进一步详细描述或参考。

[0117] 在1010中,例如,处理器1302可以输出与第一扫描单元相对应的第一位置和与第二扫描单元相对应的第二位置。在一些实施例中,如果每个1008有不超过三个连续扫描单元被定位为缺陷,则CUT可以被评估为能够进行PFA,以准备进行更严格的测试和形式验证。这样的连续扫描单元可以基于处理器1302在1010个位置中输出多个缺陷来确定。位置可以作为适合在EDA应用中使用或经由EDA工具管道中的其他工具使用的表示(例如,数据结构、文本报告等)被输出,这些工具可以通过各种协议、API或其他等效方式进行接口。图2-5提供了根据一些实施例的各种输出的一些非限制性示例。用于测试电子电路(例如,CUT)结构的模拟输出的其他示例和进一步细节在本文中的其他地方进一步详细描述或参考。

[0118] 上面进一步描述了处理器1302如何执行方法1000的任何操作或步骤的示例,诸如关于图1-13。在一些实施例中,以上步骤中的任何或所有步骤可以作为如上文关于图1-13进一步示出和描述的实施例的一部分来执行。另外地或替代地,例如,任何或所有上述步骤可以作为图2-10中所示的处理的一部分来执行。

[0119] 并非在所有情况下都需要过程1000的所有步骤来执行本文中公开的增强技术。此外,过程1000的一些步骤可以同时执行,或者以与图10所示的不同的顺序执行,如本领域普通技术人员将理解的。

[0120] 图11示出了在诸如集成电路等制品的设计、验证和制造期间使用以转换和验证表

示集成电路的设计数据和指令的一组示例过程1100。这些过程中的每个可以作为多个模块或操作来构建和启用。术语“EDA”表示术语“电子设计自动化”。这些过程开始于使用由设计者提供的信息创建产品创意1110,该信息被转换以创建使用一组EDA过程的制品1112。当设计完成时,设计被流片1134,即当集成电路的艺术品(例如,几何图案)被发送到制造设施以制造掩模组时,掩模组然后被用于制造集成电路。在流片之后,制造半导体管芯1136并且执行封装和组装过程1138以生产成品集成电路1140。

[0121] 电路或电子结构的规范的范围可以从低级晶体管材料布局到高级描述语言。高级抽象可以用于使用硬件描述语言(‘HDL’),诸如VHDL、Verilog、SystemVerilog、SystemC、MyHDL或OpenVera,来设计电路和系统。HDL描述可以转换为逻辑级别寄存器传输级(‘RTL’)描述、门级描述、布局级描述或掩模级描述。每个较低抽象级别的抽象程度较低的描述都会向设计描述中添加更多有用细节,例如,包括该描述的模块的更多细节。较低抽象级别的抽象描述可以由计算机生成,从设计库派生,或由另一设计自动化过程创建。用于指定更详细描述的低级抽象级别的规范语言的示例是SPICE,它用于对具有很多模拟组件的电路进行详细描述。每个抽象级别的描述都可以由该层的对应工具(例如,形式验证工具)使用。设计过程可以使用图11中描绘的序列。所描述的过程由EDA产品(或工具)启用。

[0122] 在系统设计1114期间,指定要制造的集成电路的功能。可以针对期望特性优化设计,诸如功耗、性能、面积(物理和/或代码行)和成本降低等。在该阶段可以将设计划分为不同类型的模块或组件。

[0123] 在逻辑设计和功能验证1116期间,用一种或多种描述语言指定电路中的模块或组件,并且检查规格的功能准确性。例如,可以验证电路的组件以生成与正在设计的电路或系统的规范要求相匹配的输出。功能验证可以使用仿真器和其他程序,诸如测试台生成器、静态HDL检查器和形式验证器。在一些实施例中,被称为“仿真器”或“原型系统”的特殊组件系统用于加速功能验证。

[0124] 在测试的合成和设计1118期间,HDL代码被转换为网表。在一些实施例中,网表可以是图结构,其中图结构的边表示电路的组件并且其中图结构的节点表示组件如何互连。HDL代码和网表都是分层制品,EDA产品可以使用该分层制品来验证集成电路在制造时是否根据指定设计执行。可以针对目标半导体制造技术优化网表。此外,可以测试成品集成电路以验证集成电路是否满足规范的要求。

[0125] 在网表验证1120期间,检查网表是否符合时序约束并且与HDL代码对应。在设计规划1122期间,针对时序和顶层布线构建和分析集成电路的总体平面图。

[0126] 在布局或物理实现1124期间,发生物理布局(定位诸如晶体管或电容器等电路组件)和布线(通过多个导体连接电路组件),并且可以执行从库中选择单元以启用特定逻辑功能。如本文中使用的,术语“单元”可以指定提供布尔逻辑功能(例如,AND、OR、NOT、XOR)或存储功能(诸如触发器或锁存器)的一组晶体管、其他组件和互连。如本文中使用的,电路“块”可以是指两个或更多个单元。单元和电路块都可以称为模块或组件,并且既可以作为物理结构也可以在模拟中使用。为所选择的单元格(基于“标准单元”)指定参数,例如大小,并且可以在数据库中访问参数以供EDA产品使用。

[0127] 在分析和提取1126期间,电路功能在布局级别被验证,这允许布局设计的细化。在物理验证1128期间,检查布局设计以确保制造约束是正确的,诸如DRC约束、电气约束、光刻

约束,并且电路系统功能与HDL设计规范相匹配。在分辨率增强1130期间,布局的几何形状被转换以改进电路设计的制造方式。

[0128] 在流片期间,创建数据以用于生产光刻掩模(如果适用,在应用光刻增强之后)。在掩模数据准备1132期间,“流片”数据用于生产用以生产成品集成电路的光刻掩模。

[0129] 计算机系统(例如,图13的计算机系统1300或图12的主机系统1207)的存储子系统可以用于存储程序和数据结构,该程序和数据结构由本文中描述的一些或全部EDA产品以及用于开发库单元以及使用库的物理和逻辑设计的产品使用。

[0130] 图12描绘了示例仿真环境1200的抽象图。仿真环境1200可以被配置为验证电路设计的功能。仿真环境1200可以包括主机系统1207(例如,作为EDA系统一部分的计算机)和仿真系统1202(例如,诸如现场可编程门阵列(FPGA)或处理器等一组可编程器件)。主机系统通过使用编译器1210构造仿真系统以仿真电路设计来生成数据和信息。要仿真的电路设计可以是指CUT,也可以称为被测设计(“DUT”),其中来自仿真的数据和信息用于验证CUT的功能。

[0131] 主机系统1207可以包括一个或多个处理器。在主机系统包括多个处理器的实施例中,本文中描述的由主机系统执行的功能可以分布在多个处理器之间。主机系统1207可以包括编译器1210以转换以表示CUT的描述语言编写的规范并且产生用于构造仿真系统1202以仿真CUT的数据(例如,二进制数据)和信息。编译器1210可以转换CUT、改变CUT、重构CUT、向CUT添加新功能、和/或控制CUT的时序。

[0132] 主机系统1207和仿真系统1202使用由仿真连接承载的信号交换数据和信息。该连接可以是但不限于一根或多根电缆,诸如具有与推荐标准232(RS232)或通用串行总线(USB)协议兼容的引脚结构的电缆。该连接可以是有线通信介质或网络,诸如局域网或广域网,诸如互联网。该连接可以是无线通信介质或具有一个或多个访问点的网络,该网络使用诸如Bluetooth或IEEE 1202.11等无线协议。主机系统1207和仿真系统1202可以通过诸如网络服务器等第三设备交换数据和信息。

[0133] 仿真系统1202包括多个FPGA(或其他模块),诸如FPGA 1204₁和1204₂以及附加FPGA 1204_N。每个FPGA可以包括一个或多个FPGA接口,FPGA通过这些接口连接到其他FPGA(以及潜在的其他仿真组件),以便FPGA交换信号。FPGA接口可以称为输入/输出引脚或FPGA焊盘。虽然仿真器可以包括FPGA,但是仿真器的实施例可以包括其他类型的逻辑块来代替或补充用于仿真CUT的FPGA。例如,仿真系统1202可以包括定制FPGA、用于仿真或原型设计的专用ASIC、存储器、和输入/输出设备。

[0134] 可编程器件可以包括可编程逻辑块阵列和互连层级,该互连层级使得可编程逻辑块能够根据HDL代码中的描述互连。每个可编程逻辑块可以启用复杂组合功能或启用逻辑门,诸如AND和XOR逻辑块。在一些实施例中,逻辑块还可以包括存储器元件/设备,该存储器元件/设备可以是简单锁存器、触发器或其他存储器块。根据不同逻辑块之间的互连的长度,信号可以在不同时间到达逻辑块的输入端子,并且因此可以暂时存储在存储器元件/设备中。

[0135] FPGA 1204₁-804_N可以放置在一个或多个板1212₁和1212₂以及附加板1212_M上。多个板可以放置到仿真单元1214₁中。可以使用仿真单元的背板或任何其他类型的连接来连接仿真单元内的板。此外,多个仿真单元(例如,1214₁和1214₂到1214_K)可以通过电缆或任何其

他方式相互连接,以形成多仿真单元系统。

[0136] 对于要仿真的CUT,主机系统300向仿真系统1202传输一个或多个位文件。位文件可以指定CUT的描述并且可以进一步指定由主机系统1207创建的CUT的分区,该分区具有跟踪和注入逻辑、分区到仿真器的FPGA的映射、以及设计约束。使用位文件,仿真器构建FPGA以执行CUT的功能。在一些实施例中,仿真器的一个或多个FPGA可以具有内置到FPGA的硅中的跟踪和注入逻辑。在这样的实施例中,FPGA可以不被主机系统构造为仿真跟踪和注入逻辑。

[0137] 主机系统1207接收要仿真的CUT的描述。在一些实施例中,CUT描述采用描述语言(例如,寄存器传输语言(RTL))。在一些实施例中,CUT描述在网表级文件或网表级文件和HDL文件的混合中。如果部分CUT描述或整个CUT描述在HDL中,则主机系统可以合成CUT描述以使用CUT描述创建门级网表。主机系统可以使用CUT的网表将CUT划分为多个分区,其中一个或多个分区包括跟踪和注入逻辑。跟踪和注入逻辑跟踪经由FPGA的接口交换的接口信号。此外,跟踪和注入逻辑可以将所跟踪的接口信号注入到FPGA的逻辑中。主机系统将每个分区映射到仿真器的FPGA。在一些实施例中,跟踪和注入逻辑被包括在一组FPGA的选择分区中。跟踪和注入逻辑可以内置到仿真器的一个或多个FPGA中。主机系统可以合成要被映射到FPGA的多路复用器。跟踪和注入逻辑可以使用多路复用器将接口信号注入到CUT逻辑中。

[0138] 主机系统创建位文件,该文件描述CUT的每个分区和分区到FPGA的映射。对于其中包括跟踪和注入逻辑的分区,位文件还描述所包括的逻辑。位文件可以包括布局和布线信息以及设计约束。主机系统存储描述哪些FPGA将仿真CUT的每个组件(例如,每个组件映射到哪些FPGA)的位文件和信息。

[0139] 根据请求,主机系统将位文件传输到仿真器。主机系统发信号通知仿真器启动CUT的仿真。在CUT的仿真期间或在仿真结束时,主机系统通过仿真连接从仿真器接收仿真结果。仿真结果是由仿真器在CUT的仿真期间生成的数据和信息,包括每个FPGA的跟踪和注入逻辑已经跟踪的接口信号和接口信号的状态。主机系统可以存储仿真结果和/或将仿真结果传输到另一处理系统。

[0140] 在CUT的仿真之后,电路设计者可以请求调试CUT的组件。如果做出这样的请求,电路设计者可以指定要调试的仿真的时间段。主机系统使用所存储的信息标识哪些FPGA正在仿真组件。主机系统检索与该时间段相关的并且由每个标识出的FPGA的跟踪和注入逻辑跟踪的所存储的接口信号。主机系统发信号通知仿真器重新仿真标识出的FPGA。主机系统将检索到的接口信号传输到仿真器,以在指定时间段内重新仿真组件。每个标识出的FPGA的跟踪和注入逻辑将从主机系统接收的其相应接口信号注入到被映射到FPGA的CUT的逻辑中。在对FPGA进行多次重新仿真的情况下,合并结果产生完整的调试视图。

[0141] 主机系统从仿真系统接收在组件的重新仿真期间由所标识出的FPGA的逻辑跟踪的信号。主机系统存储从仿真器接收的信号。在重新仿真期间跟踪的信号可以具有比在初始仿真期间更高的采样速率。例如,在初始仿真中,跟踪信号可以包括每X毫秒保存的组件状态。然而,在重新仿真中,跟踪信号可以包括每Y毫秒保存的状态,其中Y小于X。如果电路设计者请求查看在重新仿真期间跟踪的信号的波形,则主机系统可以检索所存储的信号并且显示信号图。例如,主机系统可以生成信号的波形。之后,电路设计者可以请求在不同时

间段内重新仿真同一组件或者重新仿真另一组件。

[0142] 主机系统1207和/或编译器1210可以包括子系统,诸如但不限于设计合成器子系统、映射子系统、运行时子系统、结果子系统、调试子系统、波形子系统和存储子系统。子系统可以被构造和启用为个体或多个模块,或者两个或更多个可以被构造为模块。这些子系统共同构成仿真器并且监测仿真结果。

[0143] 设计合成器子系统将表示CUT 1205的HDL转换为门级逻辑。对于要仿真的CUT,设计合成器子系统接收CUT的描述。如果CUT的描述全部或部分在HDL中(例如,RTL或其他抽象级别),则设计合成器子系统合成CUT的HDL以创建具有在门级逻辑方面的CUT描述的门级网表。

[0144] 映射子系统对CUT进行分区并且将这些分区映射到仿真器FPGA。映射子系统使用CUT的网表将门级CUT划分为多个分区。对于每个分区,映射子系统检索跟踪和注入逻辑的门级描述并且将逻辑添加到分区。如上所述,分区中包括的跟踪和注入逻辑用于跟踪经由分区被映射到的FPGA的接口交换的信号(跟踪接口信号)。可以在分区之前将跟踪和注入逻辑添加到CUT。例如,设计合成器子系统可以在合成CUT的HDL之前或之后添加跟踪和注入逻辑。

[0145] 除了包括跟踪和注入逻辑,映射子系统可以在分区中包括附加跟踪逻辑以跟踪未被跟踪和注入跟踪的某些CUT组件的状态。映射子系统可以在分区之前在CUT中或在分区之后在分区中包括附加跟踪逻辑。设计合成器子系统可以在合成HDL描述之前在CUT的HDL描述中包括附加跟踪逻辑。

[0146] 映射子系统将CUT的每个分区映射到仿真器的FPGA。对于分区和映射,映射子系统使用设计规则、设计约束(例如,时序或逻辑约束)和有关仿真器的信息。对于CUT的组件,映射子系统在存储子系统中存储描述哪些FPGA将仿真每个组件的信息。

[0147] 使用分区和映射,映射子系统生成一个或多个位文件,这些文件描述所创建的分区和逻辑到仿真器的每个FPGA的映射。位文件可以包括附加信息,诸如CUT的约束和FPGA之间的连接以及每个FPGA内的连接的布线信息。映射子系统可以为CUT的每个分区生成位文件,并且将该位文件存储在存储子系统中。根据电路设计者的要求,映射子系统将位文件传输到仿真器,并且仿真器可以使用位文件来构建FPGA以仿真CUT。

[0148] 如果仿真器包括包含跟踪和注入逻辑的专用ASIC,则映射子系统可以生成将专用ASIC连接到CUT的特定结构。在一些实施例中,映射子系统可以保存所跟踪/注入的信号的信息以及该信息在专用ASIC上的存储位置。

[0149] 运行时子系统控制由仿真器执行的仿真。运行时子系统可以引起仿真器开始或停止执行仿真。此外,运行时子系统可以向仿真器提供输入信号和数据。输入信号可以通过连接直接提供给仿真器,也可以通过其他输入信号设备间接提供给仿真器。例如,主机系统可以控制输入信号设备向仿真器提供输入信号。输入信号设备可以是例如测试板(直接或通过电缆)、信号发生器、另一仿真器或另一主机系统。

[0150] 结果子系统处理由仿真器生成的仿真结果。在仿真期间和/或在完成仿真之后,结果子系统从仿真期间生成的仿真器接收仿真结果。仿真结果包括在仿真期间跟踪的信号。具体地,仿真结果包括由每个FPGA仿真的跟踪和注入逻辑所跟踪的接口信号,并且可以包括由CUT中包括的附加逻辑所跟踪的信号。每个跟踪信号可以跨越仿真的多个周期。跟踪信

号包括多个状态,并且每个状态与仿真时间相关联。结果子系统将跟踪信号存储在存储子系统中。对于每个存储信号,结果子系统可以存储指示哪个FPGA生成跟踪信号的信息。

[0151] 调试子系统允许电路设计者调试CUT组件。在仿真器仿真CUT并且结果子系统在仿真期间接收到由跟踪和注入逻辑跟踪的接口信号之后,电路设计者可以请求通过在特定时间段重新仿真组件来调试CUT的组件。在调试组件的请求中,电路设计者标识组件并且指示要调试的仿真时间段。电路设计者的请求可以包括采样速率,该采样速率指示跟踪信号的逻辑应当多久保存一次调试组件的状态。

[0152] 调试子系统使用映射子系统在存储子系统中存储的信息来标识正在仿真组件的仿真器的一个或多个FPGA。对于每个所标识出的FPGA,调试子系统从存储子系统中检索在由电路设计者指示的时间段期间由FPGA的跟踪和注入逻辑跟踪的接口信号。例如,调试子系统检索由跟踪和注入逻辑跟踪的与该时间段相关联的状态。

[0153] 调试子系统将检索到的接口信号传输到仿真器。调试子系统指示调试子系统使用所标识出的FPGA,并且针对每个所标识出的FPGA的跟踪和注入逻辑,将其相应跟踪信号注入到FPGA的逻辑中以在所请求的时间段内重新仿真组件。调试子系统可以进一步将由电路设计者提供的采样速率传输到仿真器,以便跟踪逻辑以适当间隔跟踪状态。

[0154] 为了调试组件,仿真器可以使用组件已经被映射到的FPGA。此外,可以在由电路设计者指定的任何时间点执行组件的重新仿真。

[0155] 对于所标识出的FPGA,调试子系统可以向仿真器发送指令,以加载具有所标识出的FPGA的相同配置的多个仿真器FPGA。调试子系统另外发信号通知仿真器并行使用多个FPGA。来自多个FPGA的每个FPGA与接口信号的不同时间窗口一起使用,以在更短时间内生成更大时间窗口。例如,所标识出的FPGA可能需要一小时或更长时间才能使用一定量的周期。但是,如果多个FPGA具有与所标识出的FPGA相同的数据和结构,并且这些FPGA中的每个运行周期的子集,则仿真器可能需要几分钟来让FPGA共同使用所有周期。

[0156] 电路设计者可以标识层次结构或CUT信号列表以重新仿真。为了实现这一点,调试子系统确定仿真层次结构或信号列表所需要的FPGA,检索必要的接口信号,并且将检索到的接口信号传输到仿真器以进行重新仿真。因此,电路设计者可以标识要调试/重新仿真的CUT的任何元素(例如,组件、设备或信号)。

[0157] 波形子系统使用跟踪信号生成波形。如果电路设计者请求查看在仿真运行期间跟踪的信号的波形,则主机系统从存储子系统检索信号。波形子系统显示信号图。对于一个或多个信号,当从仿真器接收到信号时,波形子系统可以自动生成信号图。

[0158] 图13示出了可以在其中执行一组指令的计算机系统1300的示例机器,该组指令用于引起机器执行本文中讨论的任何一个或多个方法。在替代实现中,机器可以连接(例如,联网)到LAN、内联网、外联网和/或互联网中的其他机器。该机器可以在客户端服务器网络环境中作为服务器或客户端机器操作,在对等(或分布式)网络环境中作为对等机器操作,或者在云计算基础设施或环境中作为服务器或客户端机器操作。

[0159] 该机器可以是个人计算机(PC)、平板PC、机顶盒(STB)、个人数字助理(PDA)、蜂窝电话、网络设备、服务器、网络路由器、交换机或网桥、或能够执行一组指令(顺序或以其他方式)的任何机器,该u指令指定该机器要采取的动作。此外,虽然示出了单个机器,但术语“机器”也应当被认为包括个体或联合执行一组(或多组)指令以执行本文中讨论的任何一

个或多个方法的机器的任何集合。

[0160] 示例计算机系统1300包括处理设备1302、主存储器1304(例如,只读存储器(ROM)、闪存、动态随机存取存储器(DRAM)(诸如同步DRAM(SDRAM))、静态存储器1306(例如,闪存、静态随机存取存储器(SRAM)等)和数据存储设备1318,它们通过总线1330相互通信。

[0161] 处理设备1302表示一个或多个处理器,诸如微处理器、中央处理单元等。更具体地,处理设备可以是复杂指令集计算(CISC)微处理器、精简指令集计算(RISC)微处理器、超长指令字(VLIW)微处理器、或实现其他指令集的处理器、或实现指令集的组合的处理器。处理设备1302也可以是一个或多个专用处理设备,诸如专用集成电路(ASIC)、现场可编程门阵列(FPGA)、数字信号处理器(DSP)、网络处理器等。处理设备1302可以被配置为执行指令1326以执行本文中描述的操作和步骤。

[0162] 计算机系统1300还可以包括网络接口设备1308以通过网络1320进行通信。计算机系统1300还可以包括视频显示单元1310(例如,液晶显示器(LCD)或阴极射线管(CRT))、字母数字输入设备1312(例如,键盘)、光标控制设备1314(例如,鼠标)、图形处理单元1322、信号生成设备1316(例如,扬声器)、图形处理单元1322、视频处理单元1328和音频处理单元1332。

[0163] 数据存储设备1318可以包括机器可读存储介质1324(也称为非暂态计算机可读介质),机器可读存储介质1324上存储有体现本文中描述的方法或功能中的任何一个或多个的一组或多组指令1326或软件。指令1326还可以在其由计算机系统1300执行期间完全地或至少部分驻留在主存储器1304和/或处理设备1302内,主存储器1304和处理设备1302也构成机器可读存储介质。

[0164] 在一些实现中,指令1326包括用于实现与本公开相对应的功能的指令。虽然机器可读存储介质1324在示例实现中被示出为单个介质,但术语“机器可读存储介质”应当被视为包括存储一组或多组指令的单个介质或多个介质(例如,集中式或分布式数据库、和/或相关联的高速缓存和服务器的)。术语“机器可读存储介质”还应当被认为包括能够存储或编码由机器执行的一组指令并且引起机器和处理设备1302执行本公开的方法中的任何一个或多个指令的任何介质。因此,术语“机器可读存储介质”应当理解为包括但不限于固态存储器、光学介质和磁性介质。

[0165] 前面详细描述的一些部分已经根据对计算机存储器内的数据位进行操作的算法和符号表示来呈现。这些算法描述和表示是数据处理领域的技术人员用来最有效地向本领域其他技术人员传达他们工作的实质的方式。算法可以是导致期望结果的一系列操作。这些操作是需要对物理量进行物理操作的操作。这样的量可以采用能够被存储、组合、比较和以其他方式操纵的电或磁信号的形式。这样的信号可以被称为位、值、元素、符号、字符、项目、数字等。

[0166] 然而,应当记住,所有这些和类似术语都与适当物理量相关联,并且只是适用于这些量的方便标签。除非从本公开中清楚地另有说明,否则应当理解,在整个描述中,某些术语指的是计算机系统或类似电子计算设备的将计算机系统的寄存器和存储器中的被表示为物理(电子)量的数据操纵和转换为类似地被表示为计算机系统存储器或寄存器或其他这样的信息存储设备中的物理量的其他数据的动作和过程。

[0167] 本公开还涉及一种用于执行本文中的操作的装置。该装置可以为预期目的而专门

构造,或者它可以包括由存储在计算机中的计算机程序选择性地激活或重新配置的计算机。这种计算机程序可以存储在计算机可读存储介质中,诸如但不限于任何类型的磁盘,包括软盘、光盘、CD-ROM、磁光盘、只读存储器 (ROM)、随机存取存储器 (RAM)、EPROM、EEPROM、磁卡或光卡、或适合存储电子指令的任何类型的介质,每个都耦合到计算机系统总线。

[0168] 本文中呈现的算法和显示与任何特定计算机或其他装置没有本质上相关。各种其他系统可以与根据本文中的教导的程序一起使用,或者构造更专门的装置来执行该方法可以证明是方便的。此外,本公开没有参考任何特定编程语言进行描述。应当理解,可以使用多种编程语言来实现如本文所述的本公开的教导。

[0169] 一个或多个处理器还可以用于支持“云计算”环境中的相关操作的性能或作为“软件即服务”(SaaS)。例如,至少一些操作可以由在分布式或远程系统处可用的一组处理器来执行,这些处理器通过通信网络(例如,互联网)和通过一个或多个软件接口(例如,应用程序接口(API))可访问。

[0170] 如本文中使用的,术语“模块”表示有形的数据和信息处理设备,通常在大小和/或复杂性方面受到限制。例如,术语“模块”可以表示可以转换数据和信息的一种或多种方法或过程。术语“模块”还可以表示计算机程序中的一种或多种方法和过程的组合。术语“模块”也可以表示数字逻辑器件的小型网络,其中逻辑器件的互连为网络提供结构。包括以专用语言(诸如SystemC)指定的模块的方法和过程可以用于生成用于处理数据和信息的数字逻辑器件网络的规范,结果与从方法和过程获取的结果完全相同。

[0171] 本公开可以作为计算机程序产品或软件提供,该计算机程序产品或软件可以包括其上存储有指令的机器可读介质,该指令可以用于将计算机系统(或其他电子设备)编程为执行根据本公开的过程。机器可读介质包括用于以机器(例如,计算机)可读的形式存储信息的任何机制。例如,机器可读(例如,计算机可读)介质包括机器(例如,计算机)可读存储介质,诸如只读存储器(“ROM”)、随机存取存储器(“RAM”)、磁盘存储介质、光存储介质、闪存设备等。

[0172] 在上述公开中,已经参考本公开的具体示例实现描述了本公开的实现。很明显,在不脱离如以下权利要求中阐述的本公开的实现的更广泛的精神和范围的情况下,可以对其进行各种修改。在本公开以单数时态指代某些元素的情况下,可以在图中描绘多于一个元素,并且相似元素用相似数字标记。因此,本公开和附图被认为是说明性的而不是限制性的。

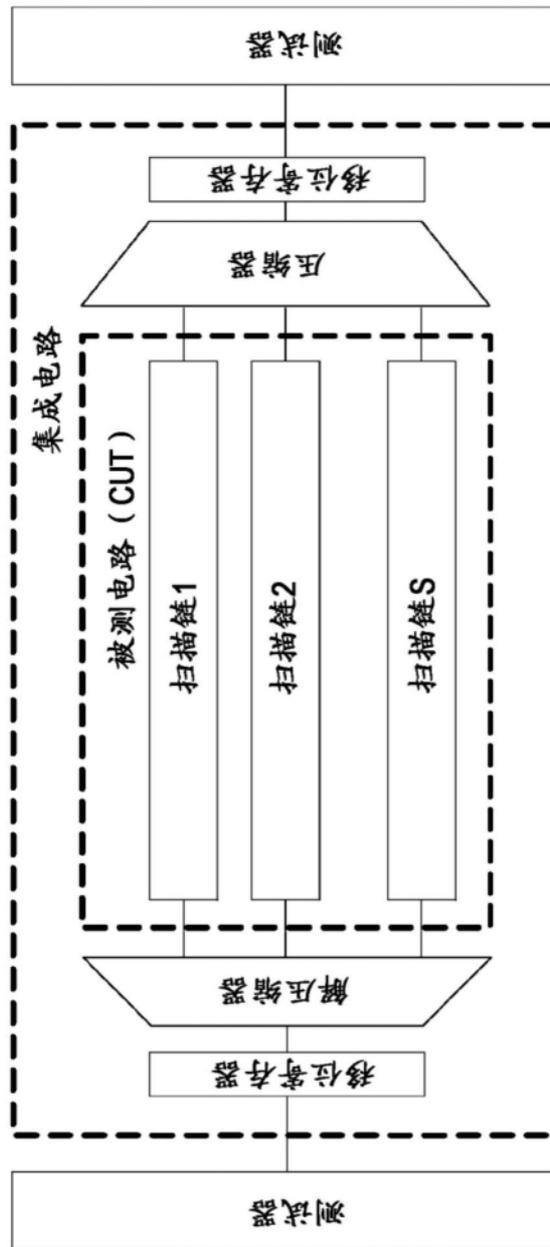


图1

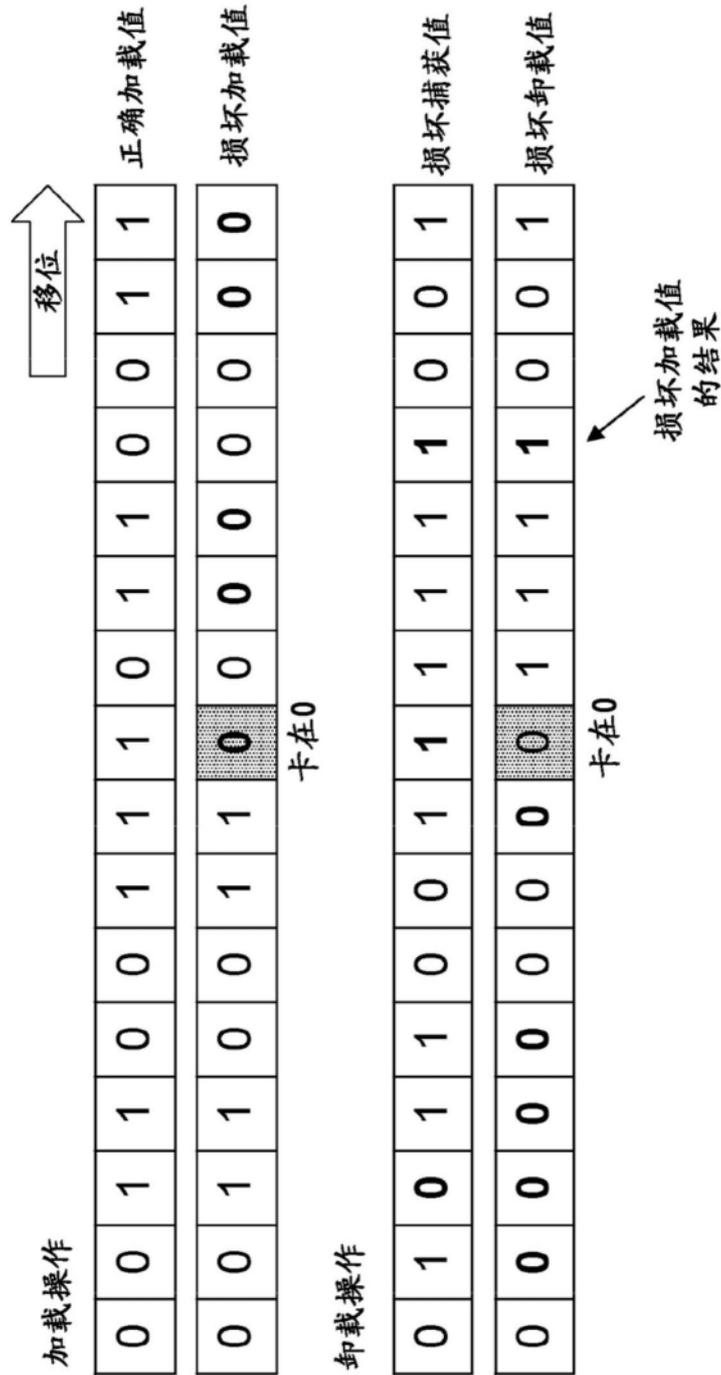


图2

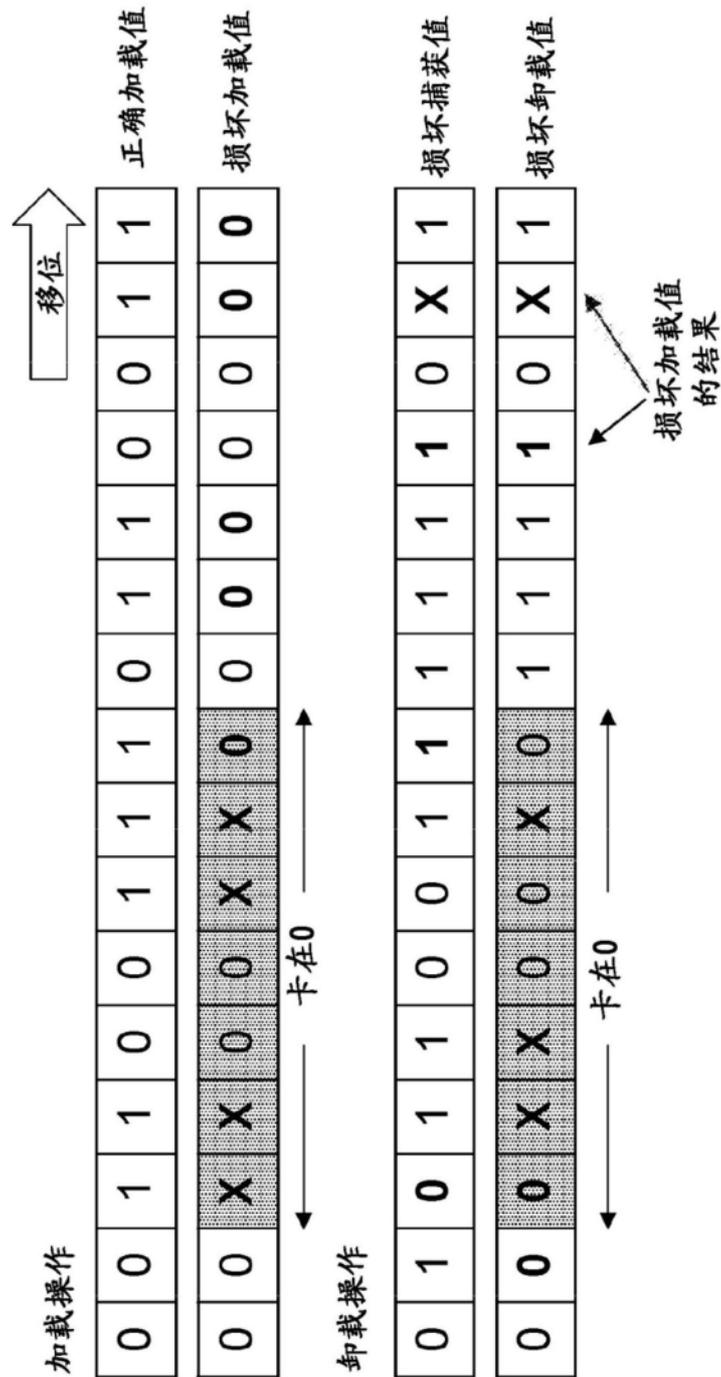


图3

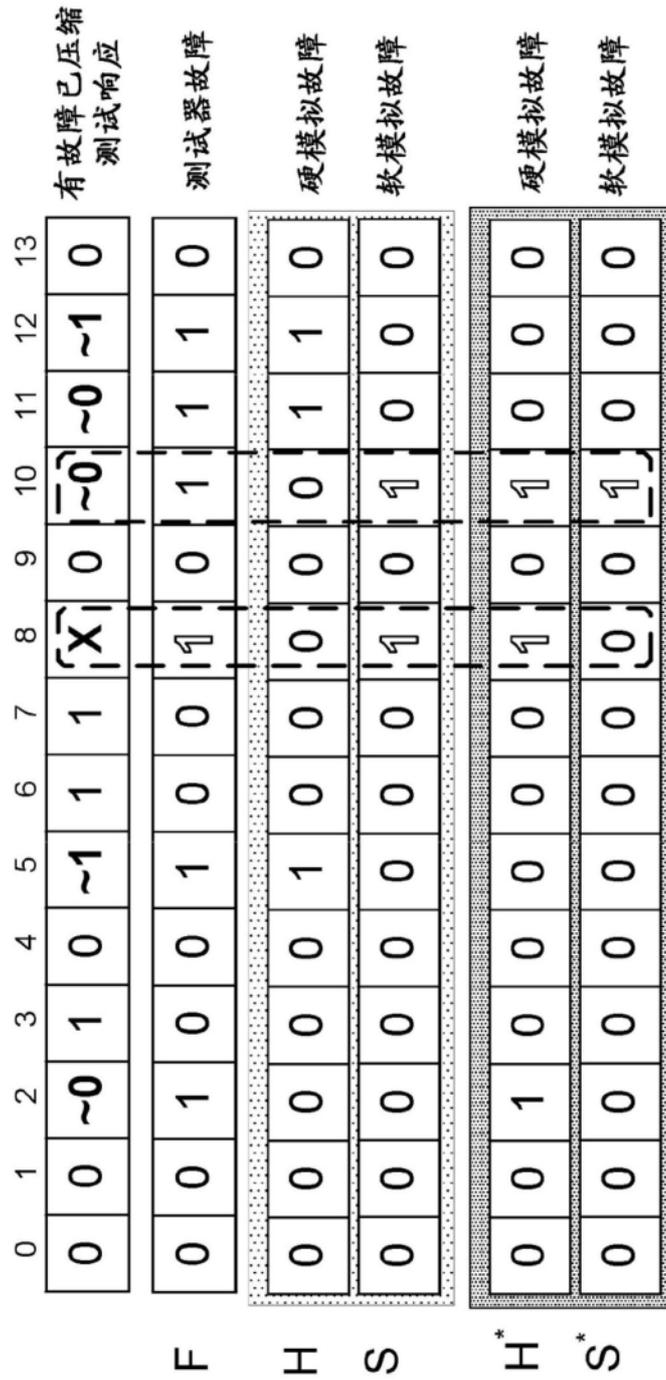


图5

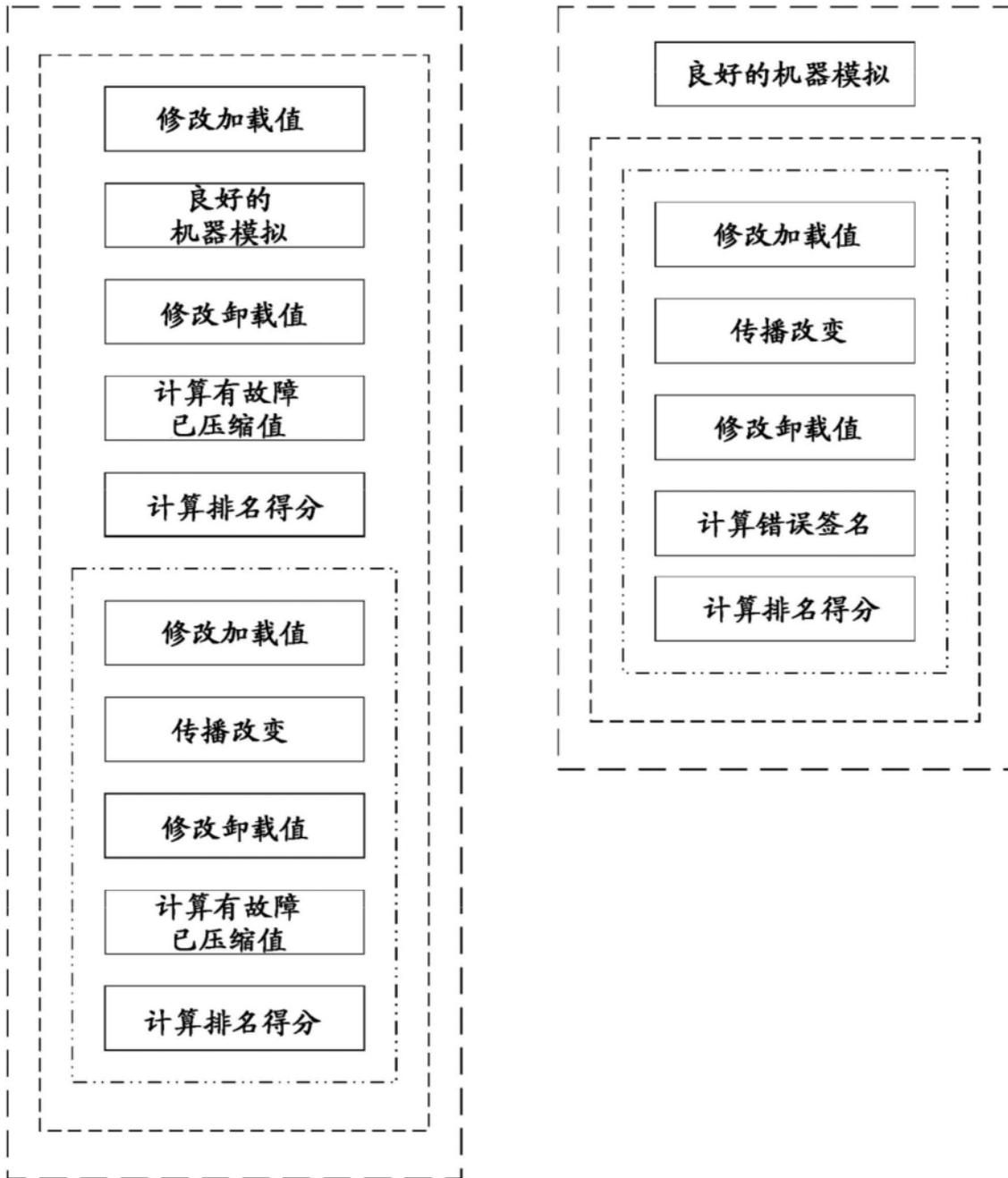


图6

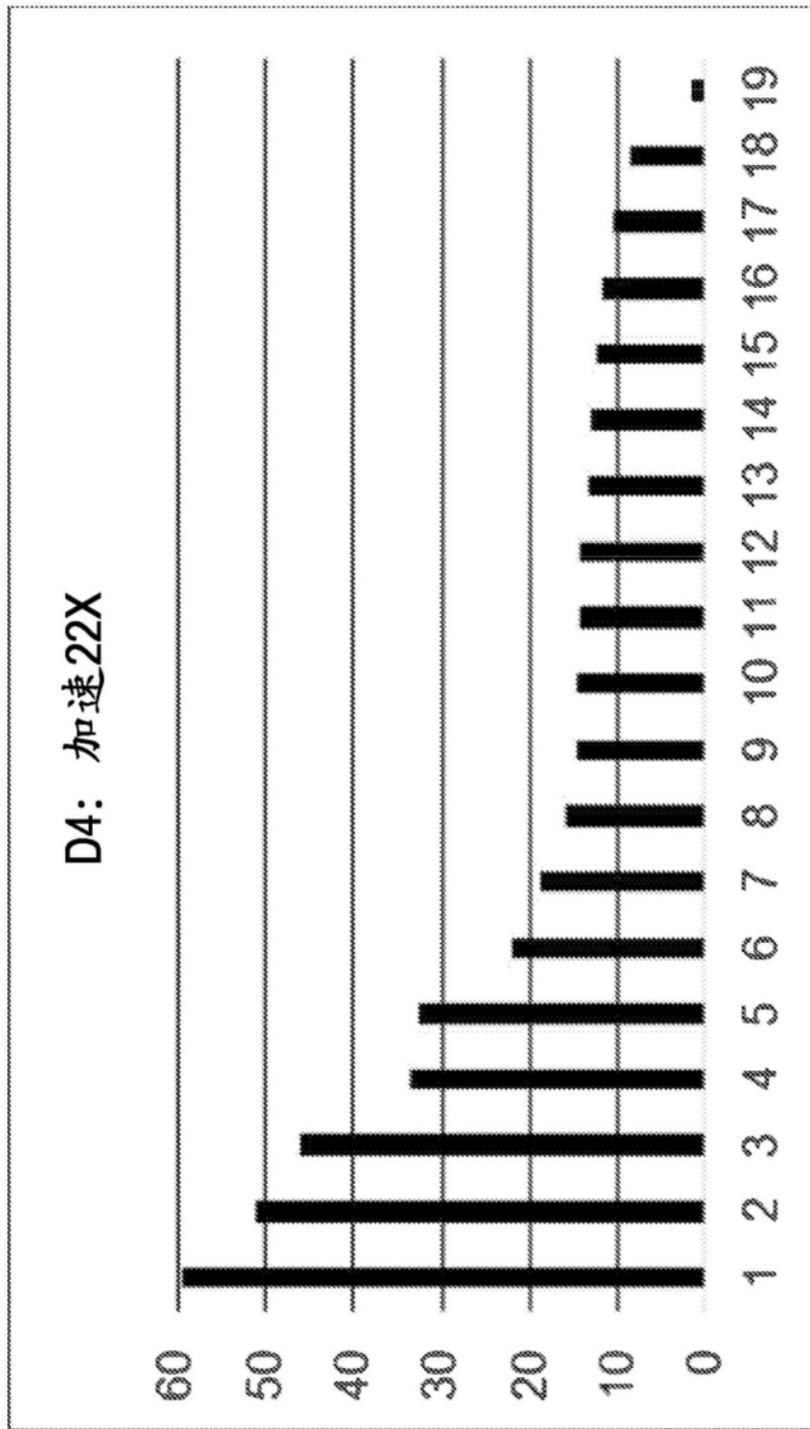


图7

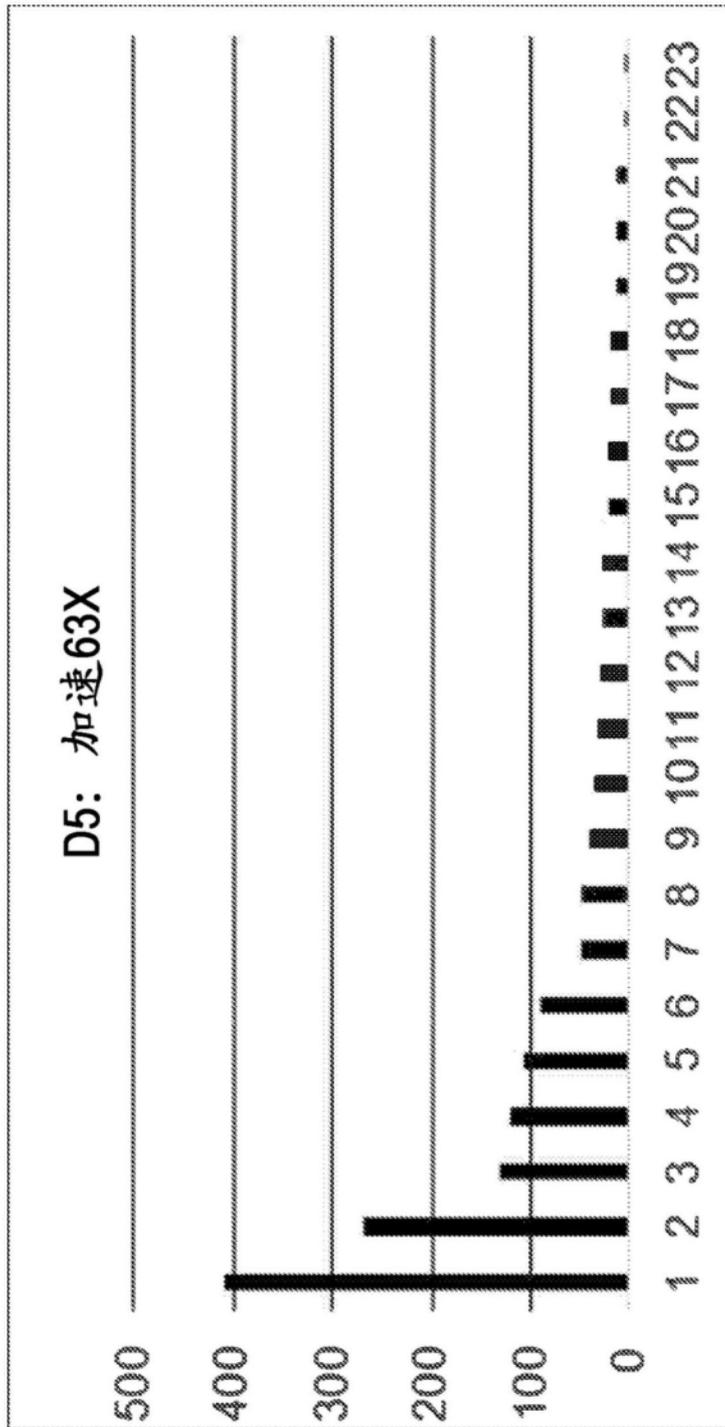


图8

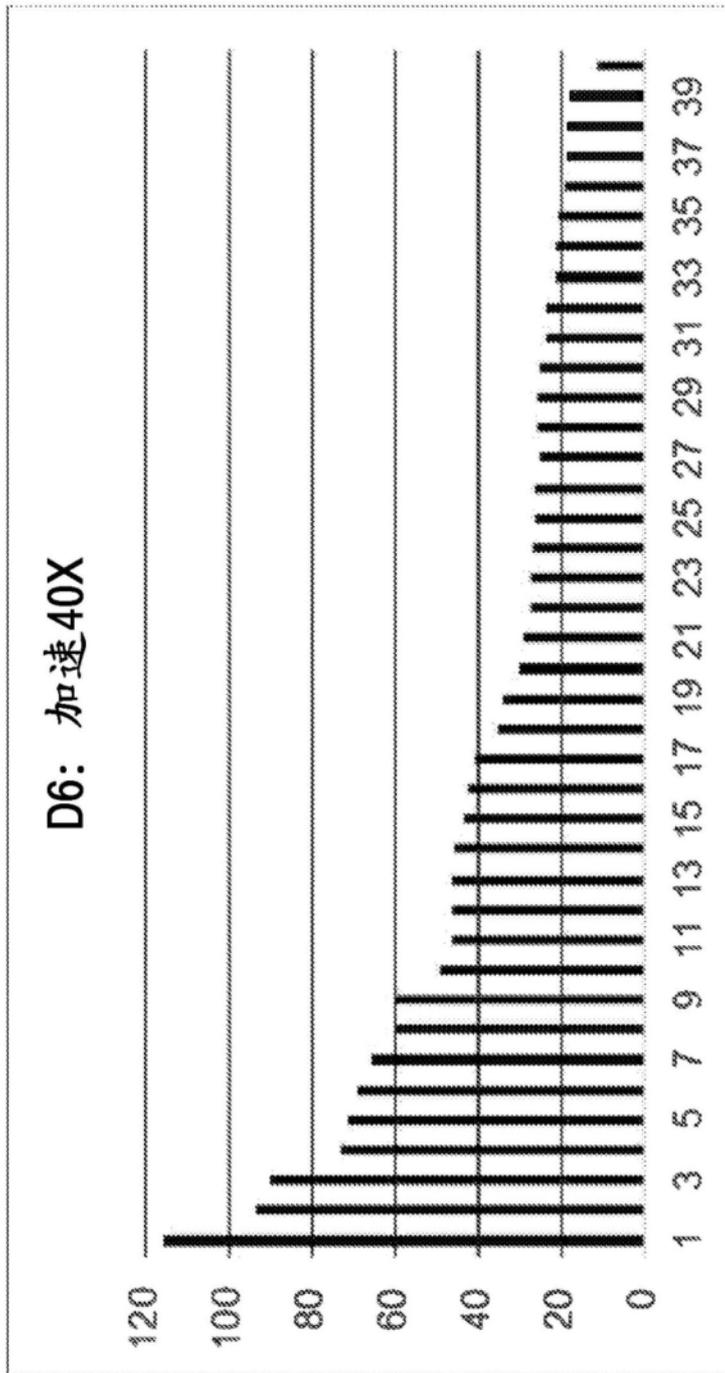


图9

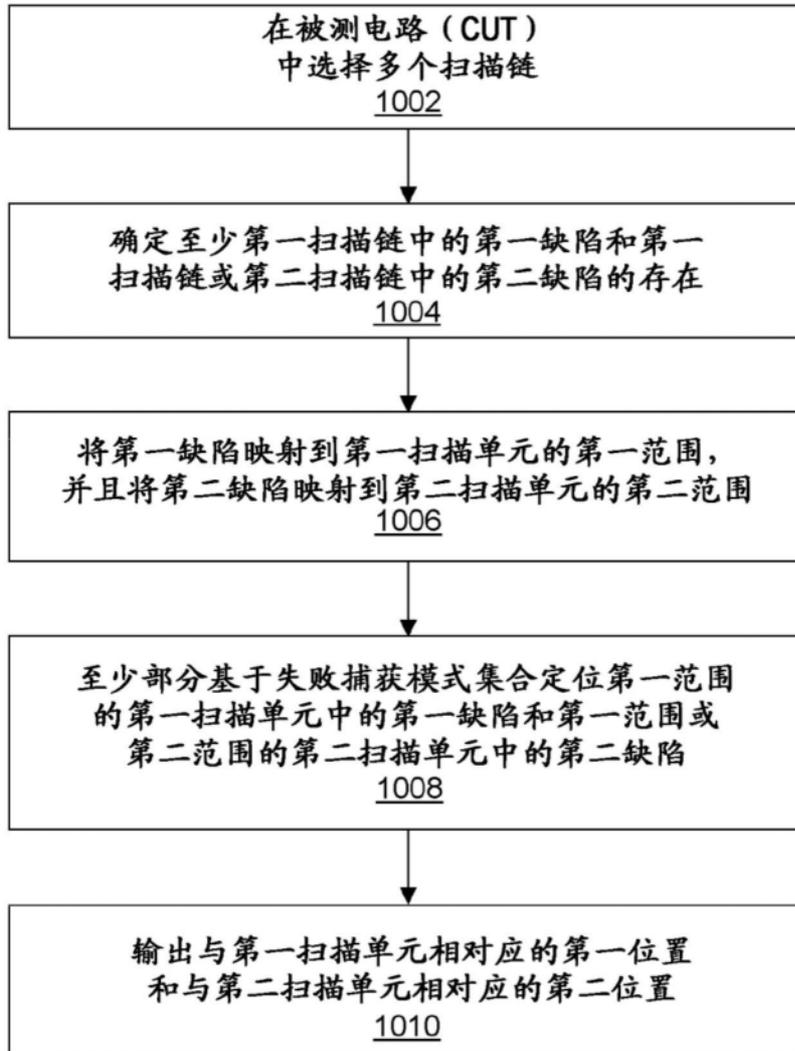
1000

图10

1100

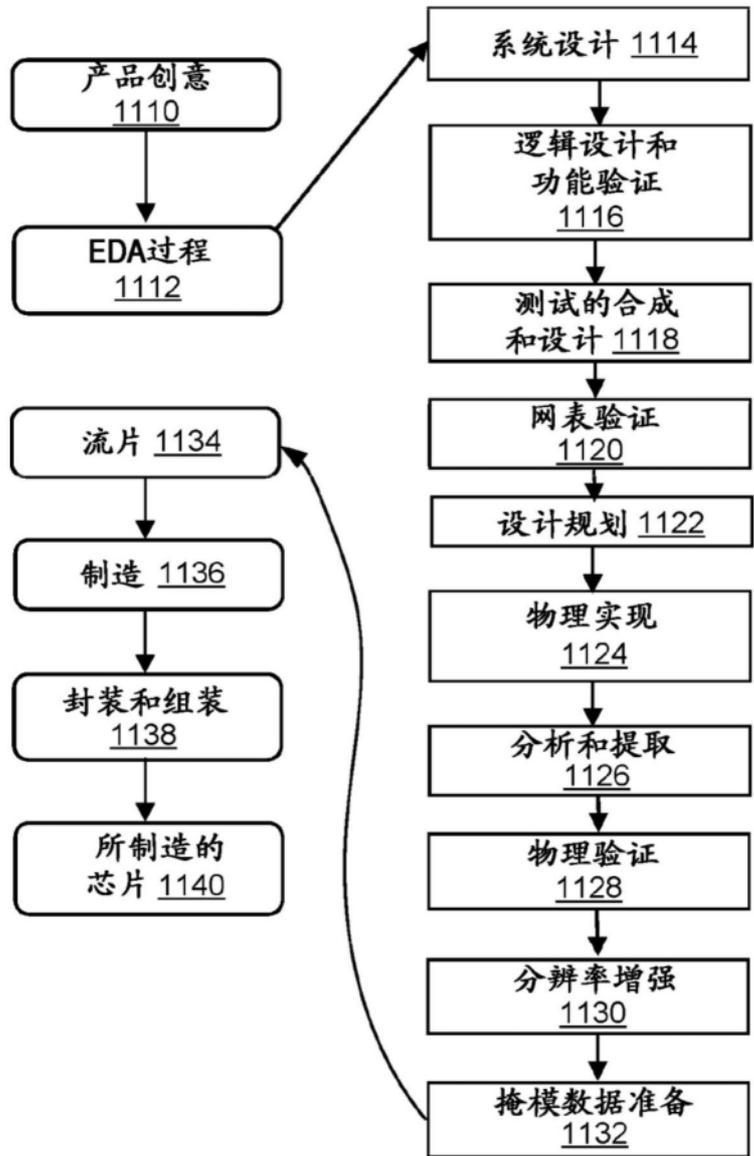


图11

1200

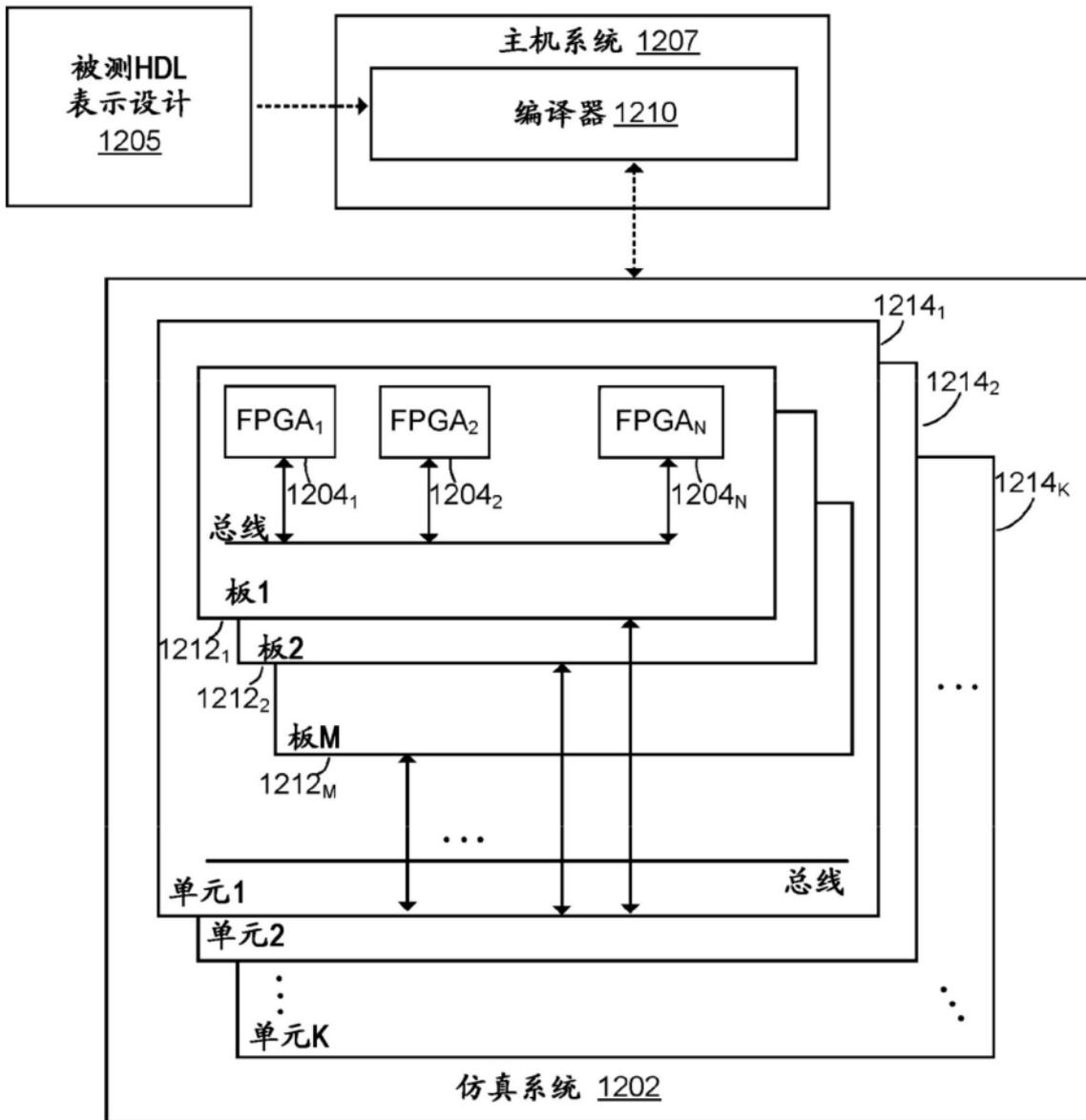


图12

1300

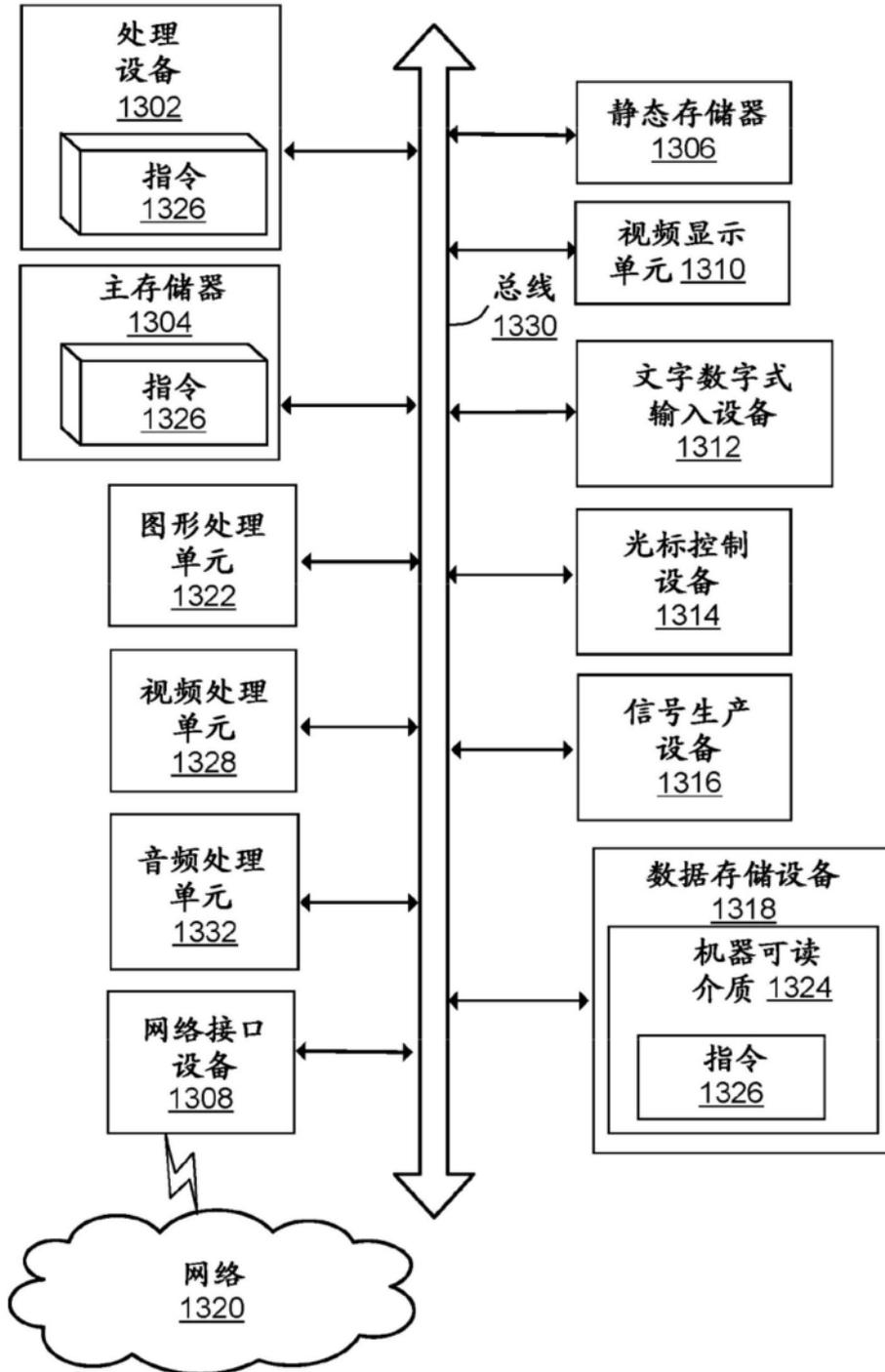


图13