



(12) 发明专利

(10) 授权公告号 CN 105893233 B

(45) 授权公告日 2021.04.27

(21) 申请号 201410813967.1

(22) 申请日 2014.12.19

(65) 同一申请的已公布的文献号
申请公布号 CN 105893233 A

(43) 申请公布日 2016.08.24

(73) 专利权人 伊姆西IP控股有限责任公司
地址 美国马萨诸塞州

(72) 发明人 杨云龙 张扬

(74) 专利代理机构 北京市金杜律师事务所
11256
代理人 王茂华 陈颖

(51) Int.Cl.
G06F 11/36 (2006.01)

(56) 对比文件

US 2013318397 A1,2013.11.28

CN 103473174 A,2013.12.25

US 6779134 B1,2004.08.17

审查员 张雪峰

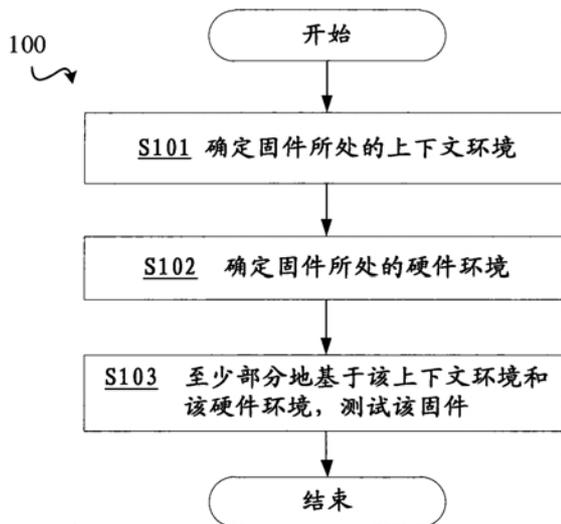
权利要求书2页 说明书7页 附图3页

(54) 发明名称

用于自动测试固件的方法和系统

(57) 摘要

本发明的实施方式提供一种用于自动测试固件的方法和系统。方法包括：确定所述固件所处的上下文环境；确定所述固件所处的硬件环境；至少部分地基于所述上下文环境和所述硬件环境，测试所述固件。通过该方法，可以在无法事先预知被测系统的实际软硬件配置的情况下自动实现对该固件的自动测试，节约了人力资源，并提高了测试效率。



1. 一种用于自动测试固件的方法,包括:
确定所述固件所处的上下文环境,其中所述上下文环境包括所述固件与何种操作系统进行接口连接;
确定所述固件所处的硬件环境;以及
至少部分地基于所述上下文环境和所述硬件环境,测试所述固件,
其中对所述固件的测试被实现为并发地执行,并且至少部分地基于所述上下文环境和所述硬件环境,测试所述固件包括:
根据所述硬件环境,修改用于所述测试的测试用例;以及
至少使用修改的所述测试用例来测试所述固件。
2. 根据权利要求1所述的方法,其中,所述至少部分地基于所述上下文环境和所述硬件环境,测试所述固件包括:
响应于所述上下文环境与预定的上下文环境不匹配,将所述固件从所述上下文环境引导至所述预定的上下文环境;以及
在所述预定的上下文环境中进行所述测试。
3. 根据权利要求1所述的方法,其中,所述硬件环境包括硬件结构和硬件接口,并且其中,所述硬件结构被抽象化,以形成用于描述所述硬件结构特性的代码,且所述硬件接口被抽象化,以形成用于描述经由所述硬件接口的通信的代码。
4. 根据权利要求3所述的方法,其中,所述硬件结构被抽象化包括:
抽象化与所述测试有关的所述硬件结构特性,而不抽象化与所述测试无关的所述硬件结构特性。
5. 一种用于自动测试多个固件的方法,包括:
为所述多个固件中的每个固件分配优先级;
至少部分地基于所述优先级为所述多个固件确定测试顺序;以及
根据所述测试顺序为所述多个固件中的每个固件执行根据权利要求1至4中任一项所述的方法。
6. 根据权利要求5所述的方法,还包括:
为用于所述多个固件中的每个固件的测试任务确定到来时间;
并且其中,所述测试顺序还基于所述到来时间而确定。
7. 根据权利要求5所述的方法,还包括:
针对所述多个固件中的当前待测试固件,确定所述当前待测试固件是否已经进入测试状态;以及
响应于所述固件已经进入所述测试状态,开始所述测试。
8. 根据权利要求7所述的方法,还包括:
响应于所述当前待测试固件尚未进入所述测试状态,跳过所述当前待测试固件,而对按照所述测试顺序的下一待测试固件确定是否已经进入所述测试状态。
9. 根据权利要求5所述的方法,其中,所述测试针对多个不同的被测平台类型或同一所述被测平台类型下的多个待测试固件而并发地进行。
10. 一种用于自动测试固件的系统,包括:
上下文确定单元,被配置为确定所述固件所处的上下文环境,其中所述上下文环境包

括所述固件与何种操作系统进行接口连接;

硬件确定单元,被配置为确定所述固件所处的硬件环境;以及

测试单元,被配置为至少部分地基于所述上下文环境和所述硬件环境,测试所述固件,其中对所述固件的测试被实现为并发地执行,并且其中所述测试单元包括:

修改单元,被配置为根据所述硬件环境,修改用于所述测试的测试用例;以及

第二子测试单元,被配置为至少使用修改的所述测试用例来测试所述固件。

11. 根据权利要求10所述的系统,其中,所述测试单元包括:

引导单元,被配置为响应于所述上下文环境与预定的上下文环境不匹配,将所述固件从所述上下文环境引导至所述预定的上下文环境;以及

第一子测试单元,被配置为在所述预定的上下文环境中进行所述测试。

12. 根据权利要求10所述的系统,其中,所述硬件环境包括硬件结构和硬件接口,并且其中,所述硬件结构被抽象化,以形成用于描述所述硬件结构特性的代码,且所述硬件接口被抽象化,以形成用于描述经由所述硬件接口的通信的代码。

13. 根据权利要求12所述的系统,其中,所述硬件结构被抽象化包括:

抽象化与所述测试有关的所述硬件结构特性,而不抽象化与所述测试无关的所述硬件结构特性。

14. 一种用于自动测试多个固件的系统,包括:

优先级分配单元,被配置为为所述多个固件中的每个固件分配优先级;

顺序确定单元,被配置为至少部分地基于所述优先级为所述多个固件确定测试顺序;

以及

执行单元,被配置为根据所述测试顺序为所述多个固件中的每个固件执行根据权利要求1至4中任一项所述的方法。

15. 根据权利要求14所述的系统,还包括:

到来时间确定单元,被配置为为用于所述多个固件中的每个固件的测试任务确定到来时间;

并且其中,所述顺序确定单元还基于所述到来时间而确定所述测试顺序。

16. 根据权利要求14所述的系统,还包括:

状态确定单元,被配置为针对所述多个固件中的当前待测试固件,确定所述当前待测试固件是否已经进入测试状态;

并且其中,所述测试单元被配置为响应于所述固件已经进入所述测试状态,开始所述测试。

17. 根据权利要求16所述的系统,其中,所述测试单元还被配置为:响应于所述当前待测试固件尚未进入所述测试状态,跳过所述当前待测试固件,而对按照所述测试顺序的下一待测试固件确定是否已经进入所述测试状态。

18. 根据权利要求14所述的系统,其中,所述测试针对多个不同的被测平台类型或同一所述被测平台类型下的多个待测试固件而并发地进行。

用于自动测试固件的方法和系统

技术领域

[0001] 本发明的实施方式涉及固件测试领域,更具体地,涉及一种用于测试固件的方法和系统。

背景技术

[0002] 随着计算机软件技术的高速发展,固件的开发和测试也成为研究的热点之一。术语“固件”一般而言是指存储于可擦除可编程只读存储器 (EPROM) 或电可擦除可编程只读存储器 (EEPROM) 等闪存芯片中的一种程序。这种程序通常担任较为基础、底层的工作。例如计算机主板上的基本输入/输出系统BIOS就是固件的一个示例。然而,随着集成电路技术的发展,固件与普通软件的界限已经不再特别明显。因此,在本文中,如无必要,“固件”与软件并不进行严格区分。

[0003] 通常,在固件的开发过程结束之后,需要测试人员来手动地对所开发的固件进行测试。这是由于被测系统状态具有不确定性,导致对固件的测试用例或步骤往往需要进行相应的调整。在缺乏有效机制支持的情况下,这种不确定性使得对固件的测试难以变为自动化。而手动测试固件本身除了耗费巨大的人力资源之外,还因为无法优化测试流程而导致效率低下。

发明内容

[0004] 为了解决现有技术中存在的上述问题,本说明书提出如下方案。

[0005] 根据本发明的第一方面,提供一种用于自动测试固件的方法,包括:确定所述固件所处的上下文环境;确定所述固件所处的硬件环境;以及至少部分地基于所述上下文环境和所述硬件环境,测试所述固件。

[0006] 在本发明的可选实施方式中,所述至少部分地基于所述上下文环境和所述硬件环境,测试所述固件包括:响应于所述上下文环境与预定的上下文环境不匹配,将所述固件从所述上下文环境引导至所述预定的上下文环境;以及在所述预定的上下文环境中进行所述测试。

[0007] 在本发明的可选实施方式中,所述至少部分地基于所述上下文环境和所述硬件环境,测试所述固件包括:根据所述硬件环境,修改用于所述测试的测试用例;以及至少使用修改的所述测试用例来测试所述固件。

[0008] 在本发明的可选实施方式中,所述硬件环境包括硬件结构和硬件接口,并且其中,所述硬件结构被抽象化,以形成用于描述所述硬件结构特性的代码,且所述硬件接口被抽象化,以形成用于描述经由所述硬件接口的通信的代码。

[0009] 在本发明的可选实施方式中,所述硬件结构被抽象化包括:抽象化与所述测试有关的所述硬件结构特性,而不抽象化与所述测试无关的所述硬件结构特性。

[0010] 根据本发明的第二方面,提供一种用于自动测试多个固件的方法,包括:为所述多个固件中的每个固件分配优先级;至少部分地基于所述优先级为所述多个固件确定测试顺

序;以及根据所述测试顺序为所述多个固件中的每个固件执行根据前述任一实施方式所述的方法。

[0011] 在本发明的可选实施方式中,该方法还包括:为用于所述多个固件中的每个固件的测试任务确定到来时间;并且其中,所述测试顺序还基于所述到来时间而确定。

[0012] 在本发明的可选实施方式中,该方法还包括:针对所述多个固件中的当前待测试固件,确定所述当前待测试固件是否已经进入测试状态;以及响应于所述固件已经进入所述测试状态,开始所述测试。

[0013] 在本发明的可选实施方式中,该方法还包括:响应于所述当前待测试固件尚未进入所述测试状态,跳过所述当前待测试固件,而对按照所述测试顺序的下一待测试固件确定是否已经进入所述测试状态。

[0014] 在本发明的可选实施方式中,所述测试针对多个不同的被测平台类型或同一所述被测平台类型下的多个待测试固件而并发地进行。

[0015] 根据本发明的第三方面,提供一种用于自动测试固件的系统,包括:上下文确定单元,被配置为确定所述固件所处的上下文环境;硬件确定单元,被配置为确定所述固件所处的硬件环境;以及测试单元,被配置为至少部分地基于所述上下文环境和所述硬件环境,测试所述固件。

[0016] 在本发明的可选实施方式中,所述测试单元包括:引导单元,被配置为响应于所述上下文环境与预定的上下文环境不匹配,将所述固件从所述上下文环境引导至所述预定的上下文环境;以及第一子测试单元,被配置为在所述预定的上下文环境中进行所述测试。

[0017] 在本发明的可选实施方式中,所述测试单元包括:修改单元,被配置为根据所述硬件环境,修改用于所述测试的测试用例;以及第二子测试单元,被配置为至少使用修改的所述测试用例来测试所述固件。

[0018] 在本发明的可选实施方式中,所述硬件环境包括硬件结构和硬件接口,并且其中,所述硬件结构被抽象化,以形成用于描述所述硬件结构特性的代码,且所述硬件接口被抽象化,以形成用于描述经由所述硬件接口的通信的代码。

[0019] 在本发明的可选实施方式中,所述硬件结构被抽象化包括:抽象化与所述测试有关的所述硬件结构特性,而不抽象化与所述测试无关的所述硬件结构特性。

[0020] 根据本发明的第四方面,提供一种用于自动测试多个固件的系统,包括:优先级分配单元,被配置为为所述多个固件中的每个固件分配优先级;顺序确定单元,被配置为至少部分地基于所述优先级为所述多个固件确定测试顺序;以及执行单元,被配置为根据所述测试顺序为所述多个固件中的每个固件执行根据权利要求1至5中任一项所述的方法。

[0021] 在本发明的可选实施方式中,该系统还包括:到来时间确定单元,被配置为为用于所述多个固件中的每个固件的测试任务确定到来时间;并且其中,所述顺序确定单元还基于所述到来时间而确定所述测试顺序。

[0022] 在本发明的可选实施方式中,该系统还包括:状态确定单元,被配置为针对所述多个固件中的当前待测试固件,确定所述当前待测试固件是否已经进入测试状态;并且其中,所述测试单元被配置为响应于所述固件已经进入所述测试状态,开始所述测试。

[0023] 在本发明的可选实施方式中,所述测试单元还被配置为:响应于所述当前待测试固件尚未进入所述测试状态,跳过所述当前待测试固件,而对按照所述测试顺序的下一待

测试固件确定是否已经进入所述测试状态。

[0024] 在本发明的可选实施方式中,所述测试针对多个不同的被测平台类型或同一所述被测平台类型下的多个待测试固件而并发地进行。

[0025] 根据本发明的实施方式,能够在无法事先预知被测系统的实际软硬件配置的情况下自动实现对该固件的自动测试,节约了人力资源,并提高了测试效率。

附图说明

[0026] 通过结合附图对本公开示例性实施方式进行更详细的描述,本公开的上述以及其它目的、特征和优势将变得更加明显,其中,在本公开示例性实施方式中,相同的参考标号通常代表相同部件。

[0027] 图1示出了根据本发明示例性实施方式的用于自动测试固件的方法100的流程图。

[0028] 图2图示了根据本发明示例性实施方式的用于自动测试多个固件的方法200的流程图;

[0029] 图3示出了根据本发明示例性实施方式的用于自动测试固件的系统300的示意性框图;

[0030] 图4示出了根据本发明示例性实施方式的用于自动测试多个固件的系统400的示意性框图;

[0031] 图5示出了适于用来实践本发明实施方式的计算机系统500的示意性框图。

具体实施方式

[0032] 下面将参照附图更详细地描述本公开的优选实施方式。虽然附图中显示了本公开的优选实施方式,然而应该理解,可以以各种形式实现本公开而不应被这里阐述的实施方式所限制。相反,提供这些实施方式是为了使本公开更加透彻和完整,并且能够将本公开的范围完整的传达给本领域的技术人员。

[0033] 图1示出了根据本发明示例性实施方式的用于自动测试固件的方法100的流程图。如图所示,方法100开始以后,首先进到步骤S101,确定该固件所处的上下文环境。需要注意的是,本文中所称的“上下文环境”主要指代相对于硬件环境而言的“软件环境”,即该固件所在的被测系统(以下简称为“被测系统”)所处的“状态”。在对固件的测试任务开始之前,首先需要了解该固件所处的上下文环境。该上下文环境包括但不限于WINDOWS操作系统、LINUX操作系统、POST、PXE操作系统、Debugger、BIOS等等。

[0034] 接着,方法100前进到步骤S102,确定该固件所处的硬件环境。如前所述,这里的“硬件环境”是相对于“上下文环境”(即“软件环境”)而言。例如,该硬件环境包括被测系统的实际硬件配置。在一个实现中,可以利用被测系统在特定状态下的行为先验知识对被测系统进行检测,根据检测结果来确定实际的硬件配置。术语“先验知识”包括但不限于通过历史行为所获知的被测系统的硬件配置信息。

[0035] 接下来,方法100前进到步骤S103,至少部分地基于该上下文环境和该硬件环境,测试该固件。根据本发明的一个实施方式,步骤S103可以包括响应于在步骤S101中确定的上下文环境与预定的上下文环境(即,目标上下文环境)不匹配,而将该固件从该上下文环境引导至该预定的上下文环境。本领域技术人员可以理解,在测试过程中,可以根据测试目

的而将被测系统引导至目标系统状态,通过检查在目标系统状态下被测系统的行为与预期结果的匹配程度来判定固件的质量。然而,本发明并不必然局限于此。此外,根据本发明的另一个实施方式,步骤S103也可以包括根据在步骤S102中所确定的硬件环境来修改用于该测试的测试用例,并且至少使用该修改的测试用例来测试该固件。例如,在实现中,一个测试案例可能包括有针对硬件环境中的各种不同硬件部件的测试片段,此时可以根据对被测系统的硬件环境检测来裁剪掉当前硬件环境中不涉及的硬件部件的相关测试片段,从而达到节约资源的效果。

[0036] 根据本发明的一个实施方式,上述硬件环境可以包括硬件结构和硬件接口。在实践中,可以基于“面向对象编程”的思想对相关硬件进行抽象化,从而实现对固件的测试。例如,可以将该硬件结构抽象化,以形成用于描述所述硬件结构特性的代码。作为示例,可以将硬件结构抽象成软件的树状结构。通过对硬件结构的抽象化,能够用代码来描述硬件配置,从而有利于通过设置代码指令来调控硬件的功能。在进一步的可选实施方式中,可以仅抽象化与固件测试有关的硬件结构特性,而不抽象化与该固件测试无关的硬件结构特性,从而进一步节约资源,提高固件测试的效率。

[0037] 备选地或附加地,也可以将该硬件接口抽象化,以形成用于描述经由所述硬件接口的通信的代码。在实践中往往借助于各种通信协议与被测系统通信,如串口、LAN等。通过对硬件接口的抽象化,可以用代码来描述经由该接口的通信,从而有利于通过设置代码指令来规划和定义接口的相关功能。

[0038] 需要注意的是,可以根据对被测系统的硬件结构和硬件接口的抽象,通过设置相关指令来收集被测系统响应于特定设置的属性和行为,从而在测试过程中能够动态地对被测系统的先验知识加以更新。而这些更新的先验知识转而又可以更为准确地如前所述用来确定被测系统的硬件环境。

[0039] 至此,方法100结束。

[0040] 可以看到,方法100通过引入对固件所处的上下文环境和硬件环境的具体考虑,能够在无法事先预知被测系统的实际软硬件配置的情况下自动实现对该固件的自动测试,节约了人力资源,并提高了测试效率。

[0041] 图2图示了根据本发明示例性实施方式的用于自动测试多个固件的方法200的流程图。

[0042] 如图2所示,方法200开始之后,首先进到步骤S201,为该多个固件中的每个固件分配优先级。在存在多个固件需要测试(即,存在针对多个固件的多个测试任务)的情况下,根据本发明的一个实施方式,可以采用为每个固件分配测试优先级的方式来协调对固件测试任务的调度。

[0043] 接下来,方法200进到步骤S202,至少部分地基于该优先级为该多个固件确定测试顺序。例如,在进行任务调度时,可以优先测试优先级较高的任务。然而,本发明并不局限于此,即,其他因素也可以用来帮助确定该多个固件的测试顺序。例如,还可以为该多个固件中的每个固件的测试任务确定到来时间,并且基于该到来时间来进一步确定该多个固件的测试顺序。

[0044] 然后,方法200前进到步骤S203,根据该测试顺序为该多个固件中的每个固件执行前述参照方法100所描述的各种方法步骤。

[0045] 根据本发明的一个实施方式,可以为每个所开发的固件设置多种状态,例如:“固件开发状态”,“准备测试状态”、“测试状态”等。其中,“固件开发状态”表示该固件正在开发过程中,并未准备好被测试。当开发完成后,可以为该“固件开发状态”设置退出标志(如,“固件可用”)来表示固件开发状态的退出,同时表示“准备测试状态”的进入。“准备测试状态”表示固件处于已经完成开发,但处于种种原因还未发出测试请求的状态。根据本发明的一个实施方式,在“准备测试状态”中,可以为每个任务分配步骤S201中所涉及的测试优先级。一旦测试请求发出,固件就处于“测试状态”,此时可以对固件进行如方法100以及200中所涉及的各种相关方法步骤。在实现中,诸如“开发状态”、“准备测试状态”,以及“测试状态”之类的各种状态可以记录在文件中供查询。在根据本发明的另一实施方式中,该文件还可以用于记载诸如步骤S201中所涉及的优先级信息等其他信息。因此,方法200可以在一个实施方式中可选地包括针对多个固件中的当前待测试固件,确定其是否已经进入测试状态,以及响应于该固件已经进入测试状态,进行该测试。在另一个实施方式中,备选地,响应于当前待测试固件尚未进入测试状态(例如处于“开发状态”或“准备测试状态”),则可以跳过该固件,而对按照测试顺序的下一待测试固件确定其是否已经进入测试状态。这样,在原本测试顺序靠前的固件尚未准备好进行测试时,方法不会阻塞于此,而是可以尝试对下一固件进行测试。

[0046] 此外,在根据本发明的另一实施方式中,还可以针对多个不同的被测平台类型或同一被测平台类型下的多个待测试固件来并发地进行测试。在一个示例性实现中,并发进行的多个测试可以在一个测试界面中呈现。

[0047] 接下来参看图3描述根据本发明示例性实施方式的用于自动测试固件的系统300。

[0048] 如图所示,系统300包括上下文确定单元301,被配置为确定所述固件所处的上下文环境;硬件确定单元302,被配置为确定所述固件所处的硬件环境;以及测试单元303,被配置为至少部分地基于所述上下文环境和所述硬件环境,测试所述固件。

[0049] 在本发明的可选实施方式中,测试单元303包括:引导单元,被配置为响应于所述上下文环境与预定的上下文环境不匹配,将所述固件从所述上下文环境引导至所述预定的上下文环境;以及第一子测试单元,被配置为在所述预定的上下文环境中进行所述测试。

[0050] 在本发明的可选实施方式中,所述测试单元303包括:修改单元,被配置为根据所述硬件环境,修改用于所述测试的测试用例;以及第二子测试单元,被配置为至少使用修改的所述测试用例来测试所述固件。

[0051] 在本发明的可选实施方式中,所述硬件环境包括硬件结构和硬件接口,并且其中,所述硬件结构被抽象化,以形成用于描述所述硬件结构特性的代码,且所述硬件接口被抽象化,以形成用于描述经由所述硬件接口的通信的代码。

[0052] 在本发明的可选实施方式中,所述硬件结构被抽象化包括:抽象化与所述测试有关的所述硬件结构特性,而不抽象化与所述测试无关的所述硬件结构特性。

[0053] 图4进一步描述根据本发明示例性实施方式的用于自动测试多个固件的系统400。如图所示,系统400包括优先级分配单元401,被配置为为所述多个固件中的每个固件分配优先级;顺序确定单元402,被配置为至少部分地基于所述优先级为所述多个固件确定测试顺序;以及执行单元403,被配置为根据所述测试顺序为所述多个固件中的每个固件执行根据前述方法100和200中任何相关方案所述的方法。

[0054] 在本发明的可选实施方式中,系统400还包括:到来时间确定单元,被配置为为用于所述多个固件中的每个固件的测试任务确定到来时间;并且其中,所述顺序确定单元还基于所述到来时间而确定所述测试顺序。

[0055] 在本发明的可选实施方式中,系统400还包括:状态确定单元,被配置为针对所述多个固件中的当前待测试固件,确定所述当前待测试固件是否已经进入测试状态;并且其中,系统300中的所述测试单元303被配置为响应于所述固件已经进入所述测试状态,开始所述测试。

[0056] 在本发明的可选实施方式中,系统300中的所述测试单元303还被配置为:响应于所述当前待测试固件尚未进入所述测试状态,跳过所述当前待测试固件,而对按照所述测试顺序的下一待测试固件确定是否已经进入所述测试状态。

[0057] 在本发明的可选实施方式中,所述测试针对多个不同的被测平台类型或同一所述被测平台类型下的多个待测试固件而并发地进行。

[0058] 下面参考图5,其示出了适于用来实践本发明实施方式的计算机系统500的示意性框图。例如,图5所示的计算机系统500可以用于实现如上文描述的用于确定应用正确性的系统300和装置400的各个部件,也可以用于固化或实现如上文描述的用于确定应用正确性的方法200的各个步骤。

[0059] 如图5所示,计算机系统可以包括:CPU(中央处理单元)501、RAM(随机存取存储器)502、ROM(只读存储器)503、系统总线504、硬盘控制器505、键盘控制器506、串行接口控制器507、并行接口控制器508、显示控制器509、硬盘510、键盘511、串行外部设备512、并行外部设备513和显示器514。在这些设备中,与系统总线504耦合的有CPU 501、RAM 502、ROM 503、硬盘控制器505、键盘控制器506、串行控制器507、并行控制器508和显示控制器509。硬盘510与硬盘控制器505耦合,键盘511与键盘控制器506耦合,串行外部设备512与串行接口控制器507耦合,并行外部设备513与并行接口控制器508耦合,以及显示器514与显示控制器509耦合。应当理解,图5所述的结构框图仅仅为了示例的目的而示出的,而不是对本发明范围的限制。在某些情况下,可以根据具体情况而增加或者减少某些设备。

[0060] 如上所述,系统300可以实现为纯硬件,例如芯片、ASIC、SOC等。这些硬件可以集成在计算机系统500中。此外,本发明的实施方式也可以通过计算机程序产品的形式实现。例如,参考图1描述的方法100和/或参考图2描述的方法200可以通过计算机程序产品来实现。该计算机程序产品可以存储在例如图5所示的RAM 504、ROM504、硬盘510和/或任何适当的存储介质中,或者通过网络从适当的位置下载到计算机系统500上。计算机程序产品可以包括计算机代码部分,其包括可由适当的处理设备(例如,图5中示出的CPU 501)执行的程序指令。所述程序指令至少可以包括用于实现方法100和/或200的步骤的指令。这些指令例如包括:用于确定所述固件所处的上下文环境的指令;用于确定所述固件所处的硬件环境的指令;以及用于至少部分地基于所述上下文环境和所述硬件环境,测试所述固件的指令。

[0061] 上文已经结合若干具体实施方式阐释了本发明的精神和原理。根据本发明的用于自动测试固件的方法和系统相对于现有技术具有诸多优点。例如,本发明通过提供适当的方法和系统来支持对固件测试的自动化实现。通过本发明提供的实施方式,可以在无法事先预知被测系统的实际软硬件配置的情况下自动实现对该固件的自动测试,节约了人力资源,并提高了测试效率。

[0062] 应当注意,本发明的实施方式可以通过硬件、软件或者软件和硬件的结合来实现。硬件部分可以利用专用逻辑来实现;软件部分可以存储在存储器中,由适当的指令执行系统,例如微处理器或者专用设计硬件来执行。本领域的普通技术人员可以理解上述的设备和方法可以使用计算机可执行指令和/或包含在处理器控制代码中来实现,例如在诸如磁盘、CD或DVD-ROM的载体介质、诸如只读存储器(固件)的可编程的存储器或者诸如光学或电子信号载体的数据载体上提供了这样的代码。本发明的设备及其模块可以由诸如超大规模集成电路或门阵列、诸如逻辑芯片、晶体管等的半导体、或者诸如现场可编程门阵列、可编程逻辑设备等的可编程硬件设备的硬件电路实现,也可以用由各种类型的处理器执行的软件实现,也可以由上述硬件电路和软件的结合例如固件来实现。

[0063] 应当注意,尽管在上文详细描述中提及了设备的若干装置或子装置,但是这种划分仅仅并非强制性的。实际上,根据本发明的实施方式,上文描述的两个或更多装置的特征和功能可以在一个装置中具体化。反之,上文描述的一个装置的特征和功能可以进一步划分为由多个装置来具体化。

[0064] 此外,尽管在附图中以特定顺序描述了本发明方法的操作,但是,这并非要求或者暗示必须按照该特定顺序来执行这些操作,或是必须执行全部所示的操作才能实现期望的结果。相反,流程图中描绘的步骤可以改变执行顺序。附加地或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,和/或将一个步骤分解为多个步骤执行。

[0065] 虽然已经参考若干具体实施方式描述了本发明,但是应该理解,本发明并不限于所公开的具体实施方式。本发明旨在涵盖所附权利要求的精神和范围内所包括的各种修改和等同布置。所附权利要求的范围符合最宽泛的解释,从而包含所有这样的修改及等同结构和功能。

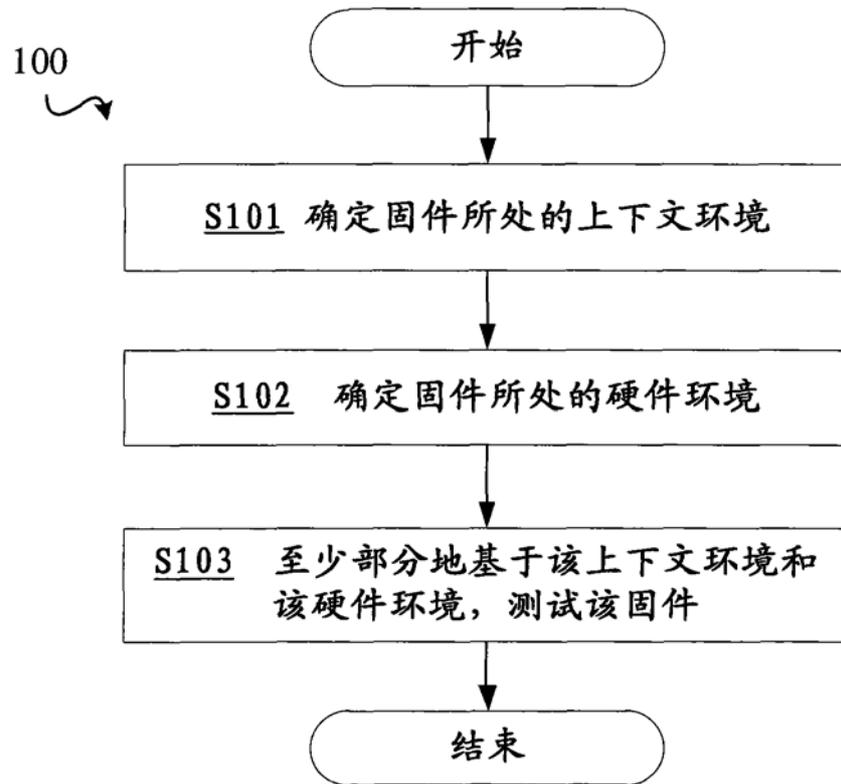


图1

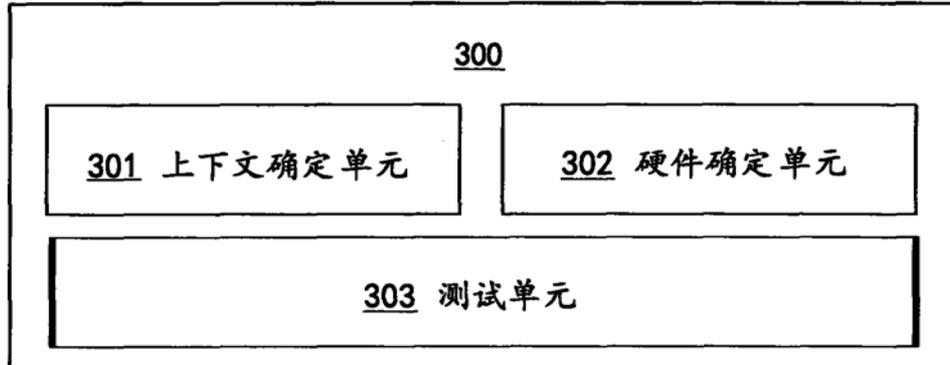


图3

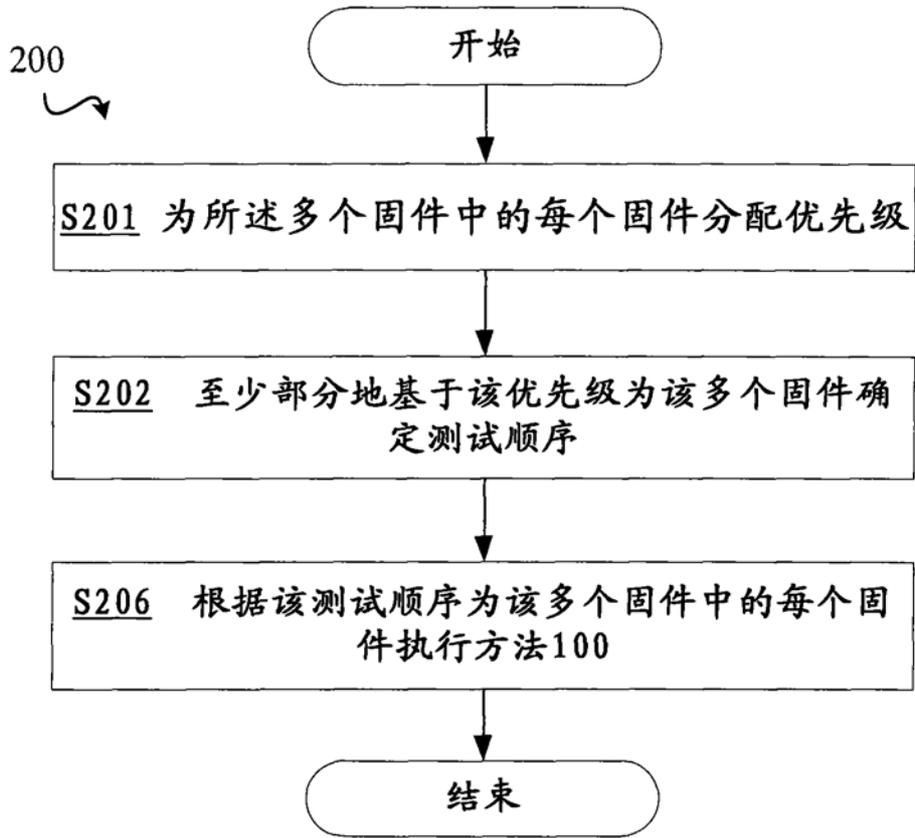


图2

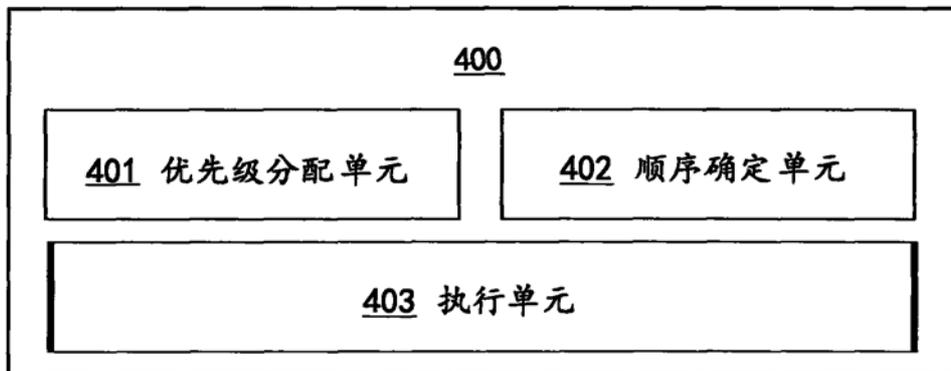


图4

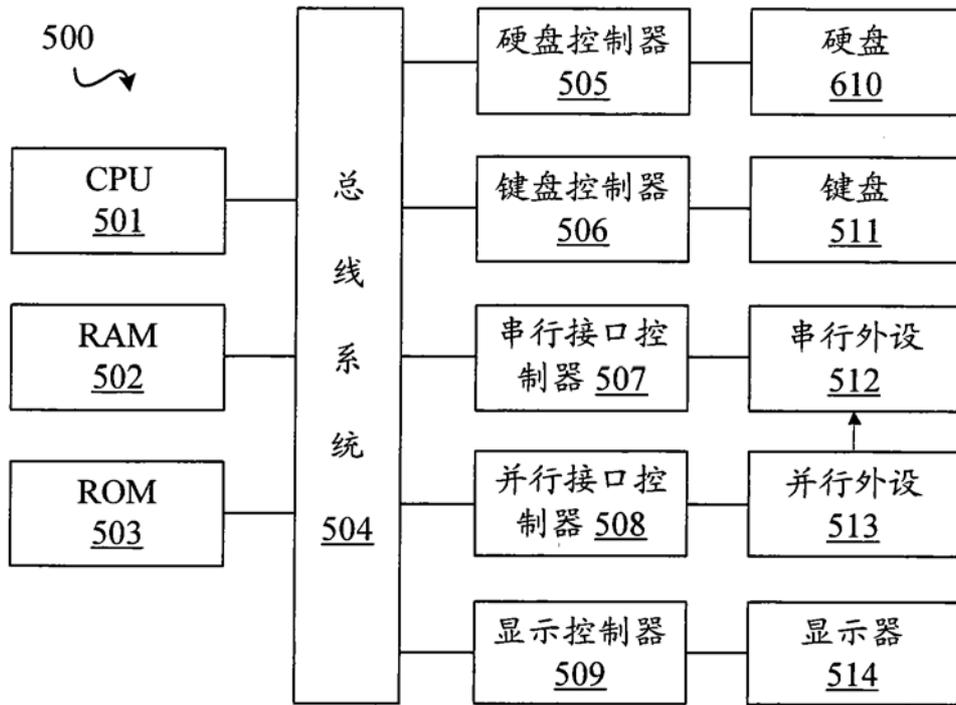


图5