



(12) 发明专利申请

(10) 申请公布号 CN 102141951 A

(43) 申请公布日 2011. 08. 03

(21) 申请号 201010560259. 3

(22) 申请日 2010. 11. 25

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
基地总部办公楼

(72) 发明人 李广

(74) 专利代理机构 北京三友知识产权代理有限
公司 11127

代理人 樊一槿

(51) Int. Cl.

G06F 11/26(2006. 01)

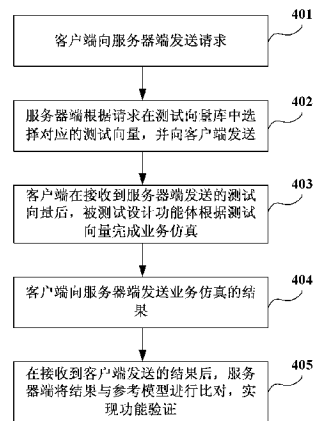
权利要求书 2 页 说明书 5 页 附图 3 页

(54) 发明名称

芯片仿真系统及方法

(57) 摘要

本发明实施例提供一种芯片仿真系统及方法,该系统包括:客户端,包括被测试设计功能体;客户端向服务器端发送请求;在接收到服务器端发送的测试向量后,被测试设计功能体根据测试向量完成业务仿真;服务器端,包括测试向量库;服务器端根据客户端的请求从测试向量库中选择对应的测试向量,并向客户端发送。通过本发明实施例,可以实现软硬件系统的解耦,提高仿真速度和仿真效率;并且,测试向量和参考模型构造简单,可以很好地支持自动化测试和多芯片仿真。



1. 一种芯片仿真系统,其特征在于,所述系统包括:

客户端,包括被测试设计功能体;所述客户端向服务器端发送请求;在接收到所述服务器端发送的测试向量后,所述被测试设计功能体根据所述测试向量完成业务仿真;

服务器端,包括测试向量库;所述服务器端根据所述客户端的请求从所述测试向量库中选择对应的测试向量,并向所述客户端发送。

2. 根据权利要求1所述的系统,其特征在于,所述服务器端还包括参考模型;

所述客户端向所述服务器端发送业务仿真的结果;

所述服务器端接收到所述结果后,将所述结果与所述参考模型进行比对,实现功能验证。

3. 根据权利要求2所述的系统,其特征在于,所述客户端具体还包括:总线功能模型、适配层和激励器;

所述适配层根据所述被测试设计功能体的请求,通过所述激励器向所述服务器端发送请求;所述激励器在接收到所述服务器端发送的测试向量后,通过所述适配层下发给所述总线功能模型;

所述总线功能模型接收到所述适配层下发的测试向量后,将所述测试向量转换成芯片接口时序并下发给所述被测试设计功能体;所述被测试设计功能体根据所述芯片接口时序进行业务仿真;以及

所述总线功能模型接收所述被测试设计功能体输出的结果后,将所述结果通过所述适配层和激励器向所述服务器端发送。

4. 根据权利要求2或3所述的系统,其特征在于,所述客户端为多个,每个客户端中的所述被测试设计功能体为芯片中的一个模块;

所述服务器端统一调度所述多个客户端的请求,实现芯片的各模块单元测试。

5. 根据权利要求2或3所述的系统,其特征在于,所述客户端为多个,每个客户端中的所述被测试设计功能体为一个芯片;

所述服务器端统一调度所述多个客户端的请求,实现多芯片的联合仿真。

6. 一种芯片仿真方法,其特征在于,所述方法包括:

客户端向服务器端发送请求;

所述服务器端根据所述请求在测试向量库中选择对应的测试向量,并向所述客户端发送;

所述客户端在接收到所述服务器端发送的测试向量后,被测试设计功能体根据所述测试向量完成业务仿真。

7. 根据权利要求6所述的方法,其特征在于,在被测试设计功能体根据所述测试向量完成业务仿真之后,所述方法还包括:

所述客户端向所述服务器端发送业务仿真的结果;

在接收到所述客户端发送的结果后,所述服务器端将所述结果与参考模型进行比对,实现功能验证。

8. 根据权利要求7所述的方法,其特征在于,所述客户端向服务器端发送请求,具体包括:所述客户端的适配层根据所述被测试设计功能体的请求,通过激励器向所述服务器端发送请求;

所述客户端在接收到所述服务器端发送的测试向量后,被测试设计功能体根据所述测试向量完成业务仿真,具体包括:所述客户端的激励器在接收到所述服务器端发送的测试向量后,通过所述适配层下发给所述总线功能模型;所述总线功能模型将所述测试向量转换成芯片接口时序并下发给所述被测试设计功能体;所述被测试设计功能体根据所述芯片接口时序进行业务仿真;

所述客户端向所述服务器端发送业务仿真的结果,具体包括:所述总线功能模型接收所述被测试设计功能体输出的结果,将所述结果通过所述适配层和激励器向所述服务器端发送。

9. 根据权利要求7所述的方法,其特征在于,所述客户端为多个,每个客户端中的所述被测试设计功能体为芯片中的一个模块;则所述服务器端统一调度所述多个客户端的请求,实现芯片的各模块单元测试;

或者每个客户端中的所述被测试设计功能体为一个芯片;则所述服务器端统一调度所述多个客户端的请求,实现多芯片的联合仿真。

10. 根据权利要求8所述的方法,其特征在于,所述总线功能模型通过可编程语言接口或者直接编程接口扩展系统函数。

芯片仿真系统及方法

技术领域

[0001] 本发明涉及大规模逻辑电路的芯片仿真领域,特别涉及一种芯片仿真系统及方法。

背景技术

[0002] 随着半导体芯片工艺水平的不断发展,芯片电路越来越复杂,经常达到百万门以上的规模;同时,对芯片质量和上市时间的要求也越来越高。这样,芯片的功能仿真就显得更加重要。

[0003] 在FPGA/ASIC等大规模逻辑电路的芯片功能仿真验证过程当中,仿真花费的时间以及仿真平台搭建的效率是一个关键问题。硬件系统逻辑电路模型通常需要与参考模型(RM, Reference Model) 或者其它芯片模型进行数据交换和通信,整个仿真系统瓶颈通常限制在仿真时间、验证向量的构建执行上。

[0004] 目前,在大规模逻辑电路功能仿真当中,主要有两种方法:一种是直接基于寄存器传输级(RTL, Register Transfer Level)进行测试环境的搭建,测试环境的构建基于硬件描述语言(HDL, Hardware Description Language)进行;

[0005] 例如,总线功能模型(BFM, Bus Function Model)、RM、激励器和被测设计功能体(DUT, Design Under Test)一样基于HDL语言进行开发,相互通信比较简单,基于HDL中的接口直接通信。

[0006] 另一种方法抽象级别比较高,使用高级语言进行建模,与DUT的通信直接使用可编程语言接口(PLI, Programming Language Interface)或者直接编程接口(DPI, Direct Programming Interface)进行通信;

[0007] 例如, RM、激励器使用高级语言进行开发;BFM基于HDL, RM、激励器与BFM的接口通过PLI或者DPI进行。

[0008] 但是在实现本发明的过程中,发明人发现现有技术的缺陷在于:第一种方法中,测试向量构造复杂, RM编写复杂;并且,由于完全基于RTL级别仿真,仿真速度慢,且自动化测试困难。

[0009] 而第二种方法中,在软硬件的联合仿真中解耦性差,由于基于集中式仿真,在大规模的芯片仿真当中,速度越来越慢,不足以支持不断增长的测试向量;另外,由于通信问题,难以支持多芯片联合仿真,仿真效率很低。

发明内容

[0010] 本发明实施例提供一种芯片仿真系统及方法,目的在于实现软硬件的解耦,提高仿真速度。

[0011] 为达到上述目的,本发明实施例提供一种芯片仿真系统,所述系统包括:

[0012] 客户端,包括被测设计功能体;客户端向服务器端发送请求;在接收到服务器端发送的测试向量后,被测设计功能体根据测试向量完成业务仿真;

[0013] 服务器端,包括测试向量库;服务器端根据客户端的请求从测试向量库中选择对应的测试向量,并向客户端发送。

[0014] 本发明实施例还提供一种芯片仿真方法,所述方法包括:客户端向服务器端发送请求;服务器端根据请求在测试向量库中选择对应的测试向量,并向客户端发送;客户端在接收到服务器端发送的测试向量后,被测试设计功能体根据所述测试向量完成业务仿真。

[0015] 本发明实施例的有益效果在于,通过将被测试设计功能体和测试向量库分布在客户端和服务器端,可以实现软硬件系统的解耦,提高仿真速度和仿真效率;并且,测试向量和参考模型构造简单,可以很好地支持自动化测试和多芯片仿真。

附图说明

[0016] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,并不构成对本发明的限定。在附图中:

[0017] 图 1 是本发明实施例的芯片仿真系统的构成示意图;

[0018] 图 2 是本发明实施例的客户端的构成示意图;

[0019] 图 3 是本发明实施例的客户端为多个的结构实例图;

[0020] 图 4 是本发明实施例的芯片仿真方法的流程图。

具体实施方式

[0021] 为使本发明的目的、技术方案和优点更加清楚明白,下面结合附图对本发明实施例作进一步详细说明。在此,本发明的示意性实施例及其说明用于解释本发明,但并不作为对本发明的限定。

[0022] 本发明实施例提供一种芯片仿真系统,如图 1 所示,所述系统包括:客户端 101 和服务器端 102;其中,

[0023] 客户端 101 包括被测试设计功能体;客户端 101 向服务器端 102 发送请求;在接收到服务器端 102 发送的测试向量后,被测试设计功能体根据测试向量完成业务仿真;

[0024] 服务器端 102 包括测试向量库;服务器端 102 根据客户端 101 的请求从测试向量库中选择对应的测试向量,并向客户端 101 发送。

[0025] 例如,客户端 101 的请求中可以包括被测试设计功能体的类型,该类型对应一组测试向量;服务器端 102 可根据该类型在测试向量库中查找到该组测试向量后,发送给客户端。但不限于此,客户端 101 的请求中还可包括组合查询条件,服务器端根据该组合查询条件选择对应的测试向量。以上仅为示意性说明,可根据实际情况确定具体的实施方式。

[0026] 进一步地,服务器端 102 还可包括参考模型。并且,在被测试设计功能体根据测试向量完成业务仿真之后,客户端 101 向服务器端 102 发送业务仿真的结果;服务器端 102 接收到该结果后,将该结果与参考模型进行比对,实现功能验证。

[0027] 在本实施例中,测试向量库中的测试向量是相互关联的,可以按照配置要求随机组合,可以非常方便的进行功能覆盖率的收集以及完善。而传统的测试方法中,每个测试向量是相对独立的,主要内容是芯片测试功能点的激励数据,各个测试向量覆盖的功能点可能存在重复;因此,影响测试效率以及功能覆盖率的收集。

[0028] 由上述实施例可知,通过将被测试设计功能体和测试向量库分布在客户端和服务端,可以实现软硬件系统的解耦,提高仿真速度和仿真效率。并且,客户端和服务端是相互独立的,可以分别由软件人员和硬件人员进行独立开发。

[0029] 具体地,如图 2 所示,客户端 101 还包括:总线功能模型 202、适配层 203 和激励器 204;其中,

[0030] 适配层 203 根据被测试设计功能体 201 的请求,通过激励器 204 向服务器端 101 发送请求;激励器 204 在接收到服务器端 102 发送的测试向量后,通过适配层 203 下发给总线功能模型 202;

[0031] 总线功能模型 202 接收到适配层 203 下发的测试向量后,将测试向量转换成芯片接口时序并下发给被测试设计功能体 201;被测试设计功能体 201 根据芯片接口时序进行业务仿真;以及

[0032] 总线功能模型 202 接收被测试设计功能体 201 输出的结果后,将结果通过适配层 203 和激励器 204 向服务器端 102 发送。

[0033] 在本实施例中,客户端 101 可以为多个,每个客户端 101 中可以有多条总线功能模型 202 和 / 或适配层 203,图 3 为本发明实施例中客户端为多个的结构实例图。如图 3 所示,服务器端可包括测试向量库和参考模型,可采用高级语言、通过软件系统实现;客户端可包括被测试设计功能体、总线功能模型、适配层、激励器和检测数据区,可通过硬件系统实现。

[0034] 在本实施例中,进一步地,每个客户端中的被测试设计功能体为芯片中的一个模块。服务器端统一调度多个客户端的请求,实现芯片的各模块单元测试。

[0035] 例如,整个芯片硬件系统可以分成多个子模块,每个子模块充当一个客户端。这样,多个客户端可以分别和服务端进行通信,由服务端分析各个客户端的请求,统一进行调度,再与其它客户端进行交互。其中,可以使用公平轮询或者带优先级权重的调度方式。

[0036] 而每个客户端可以属于网络上的任何一台主机或者服务器。这样,系统就可以快速的支持大规格芯片的各模块的单元测试,快速检测各个模块处理是否正确。

[0037] 在本实施例中,进一步地,每个客户端中的被测试设计功能体为一个芯片;服务器端统一调度多个客户端的请求,实现多芯片的联合仿真。

[0038] 例如,被测试设计功能体为一个芯片,各个芯片系统可以布局在不同的网络资源上(例如,不同主机或者服务器),通过与服务器端进行通信,由服务器端进行统一调度管理,实现芯片之间的仿真通信。这样,可以在板级验证多个芯片的处理方式是否一致,从而达到单板级别的仿真需求。

[0039] 由此,解决了多芯片仿真系统之间通信繁琐、多芯片联合仿真困难的问题,提高了芯片投片成功的机会。并且,各个芯片系统可以布局在不同的网络资源上,从而可以利用各个网络资源来加快仿真速度,克服了单一网络资源慢的问题。

[0040] 在本实施例中,进一步地,服务器端 102 可以根据网络的资源分配情况进行监控和调度,优化网络的整体运行。

[0041] 具体地,服务器端实现各个客户端的通信调度。由于各个客户端与服务端、以及其它各个客户端可能需要通信,这样,服务器端需要进行调度。可以根据整个网络的资源分配情况,进行监控和调度,从而使整个网络整体运行最优,具体算法可采用带优先级权重轮

询调度算法。

[0042] 优选地,客户端 101 进行一次初始化和配置操作,无须重复进行硬件系统初始化以及相同表项的配置。

[0043] 在本实施例中,测试用例存在于测试向量库,与客户端 101 的硬件系统是相互独立,每次运行通过参数配置从测试向量库中对测试用例进行选择执行。由于测试向量和被测设计功能体是独立的,在仿真过程当中,各个客户端的硬件系统可以一直运行,可以不随某一个测试用例执行完毕而停止。

[0044] 这样,客户端 101 中硬件系统的初始化、各种配置操作可以只进行一次,每个测试用例都可以在当前系统上执行,不用每次执行一个测试用例都需要重新下发配置和初始化。

[0045] 由此,一旦仿真系统启动之后,客户端的硬件系统可以不用停止,只需操作服务器端的测试向量库,即可不断进行测试。因此,解决了通常仿真过程当中,不同测试向量切换需要重新启动仿真环境的问题,从而减少了硬件系统初始化、配置初始化等仿真时间,可以大量节省仿真时间,提高了仿真效率。特别是在大量测试用例回归的应用当中,节省大量的回归时间,可以提高仿真效率达到 2 倍以上。

[0046] 在本实施例中,进一步地,总线功能模型 202 可以通过可编程语言接口或者直接编程接口扩展系统函数。

[0047] 例如,测试过程当中需要实时进行一些数学运算,可以通过在软件系统中快速实现。当仿真环境循环调用此系统函数的时候,作为客户端可不断向服务器端发起运算请求,服务器端通过调用软件实现的运算函数来回应。这样,可以做到随仿真时间的进行而实时响应硬件系统的运算需求。

[0048] 在本实施例中,可以使用 C/S 或者 B/S 结构,但不限于此,还可以根据实际情况采用其他结构。

[0049] 以 C/S 结构为例,服务器端可以使用 tcl,perl,python,c++/c 或者 system c 等脚本与建模语言实现,适用于测试向量的构建与软件建模。客户端适配层可以使用 c++/c 实现。客户端和服务器端之间通过 socket 套接字实现通信,既考虑到解耦性,也考虑到软件联合仿真的实际需求。

[0050] 由上述实施例可知,通过将被测试设计功能体和测试向量库分布在客户端和服务端,可以实现软硬件系统的解耦,提高仿真速度和仿真效率;并且,测试向量和参考模型构造简单,可以很好地支持自动化测试和多芯片仿真。

[0051] 本发明实施例还提供一种芯片仿真方法,如图 4 所示,所述方法包括:

[0052] 步骤 401,客户端向服务器端发送请求;

[0053] 步骤 402,服务器端根据请求在测试向量库中选择对应的测试向量,并向客户端发送;

[0054] 步骤 403,客户端在接收到服务器端发送的测试向量后,被测设计功能体根据测试向量完成业务仿真。

[0055] 进一步地,如图 4 所示,该方法还可包括:

[0056] 步骤 404,客户端向服务器端发送业务仿真的结果;

[0057] 步骤 405,在接收到客户端发送的结果后,服务器端将结果与参考模型进行比对,

实现功能验证。

[0058] 通过上述方法,可以实现软硬件系统的解耦,提高仿真速度和仿真效率。并且,客户端和服务端是相互独立的,可以分别由软件人员和硬件人员进行独立开发。

[0059] 在本实施例中,在步骤 401 实施时,具体可以包括:客户端的适配层根据被测试设计功能体的请求,通过激励器向服务器端发送请求;

[0060] 在步骤 403 实施时,具体可以包括:客户端的激励器在接收到服务器端发送的测试向量后,通过适配层下发给所述总线功能模型;

[0061] 总线功能模型将测试向量转换成芯片接口时序并下发给被测试设计功能体;被测试设计功能体根据芯片接口时序进行业务仿真。

[0062] 进一步地,总线功能模型可以不断的监听是否有指令从适配层下发,如果有,则把测试向量指令翻译成芯片接口时序关系,激发芯片的接口时序;如果没有,则继续监听。

[0063] 在步骤 404 实施时,具体可以包括:总线功能模型接收被测试设计功能体输出的结果,将结果通过适配层和激励器向服务器端发送。

[0064] 在本实施例中,进一步地,客户端为多个,每个客户端中的被测试设计功能体为芯片中的一个模块;服务器端统一调度多个客户端的请求,实现芯片的各模块单元测试。

[0065] 在本实施例中,进一步地,客户端为多个,每个客户端中的被测试设计功能体为一个芯片;服务器端统一调度多个客户端的请求,实现多芯片的联合仿真。

[0066] 在本实施例中,进一步地,服务器端根据网络的资源分配情况进行监控和调度,优化网络的整体运行。

[0067] 在本实施例中,进一步地,客户端进行一次初始化和配置操作。

[0068] 在本实施例中,进一步地,总线功能模型通过可编程语言接口或者直接编程接口扩展系统函数。

[0069] 由上述实施例可知,通过将测试设计功能体和测试向量库分布在客户端和服务端,可以实现软硬件系统的解耦,提高仿真速度和仿真效率;并且,测试向量和参考模型构造简单,可以很好地支持自动化测试和多芯片仿真。

[0070] 本领域普通技术人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0071] 结合本文中所公开的实施例描述的方法或算法的步骤可以用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器 (RAM)、内存、只读存储器 (ROM)、电可编程 ROM、电可擦除可编程 ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0072] 以上所述的具体实施方式,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施方式而已,并不用于限定本发明的保护范围,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

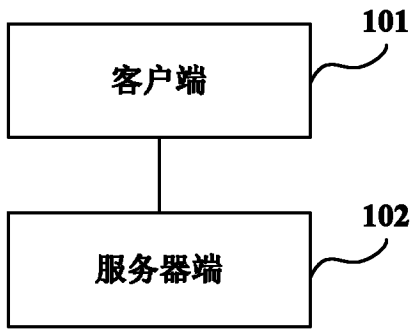


图 1

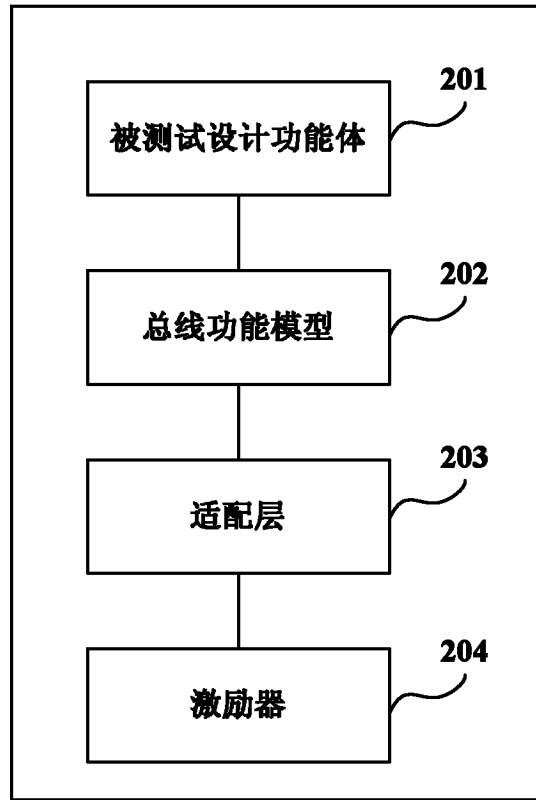


图 2

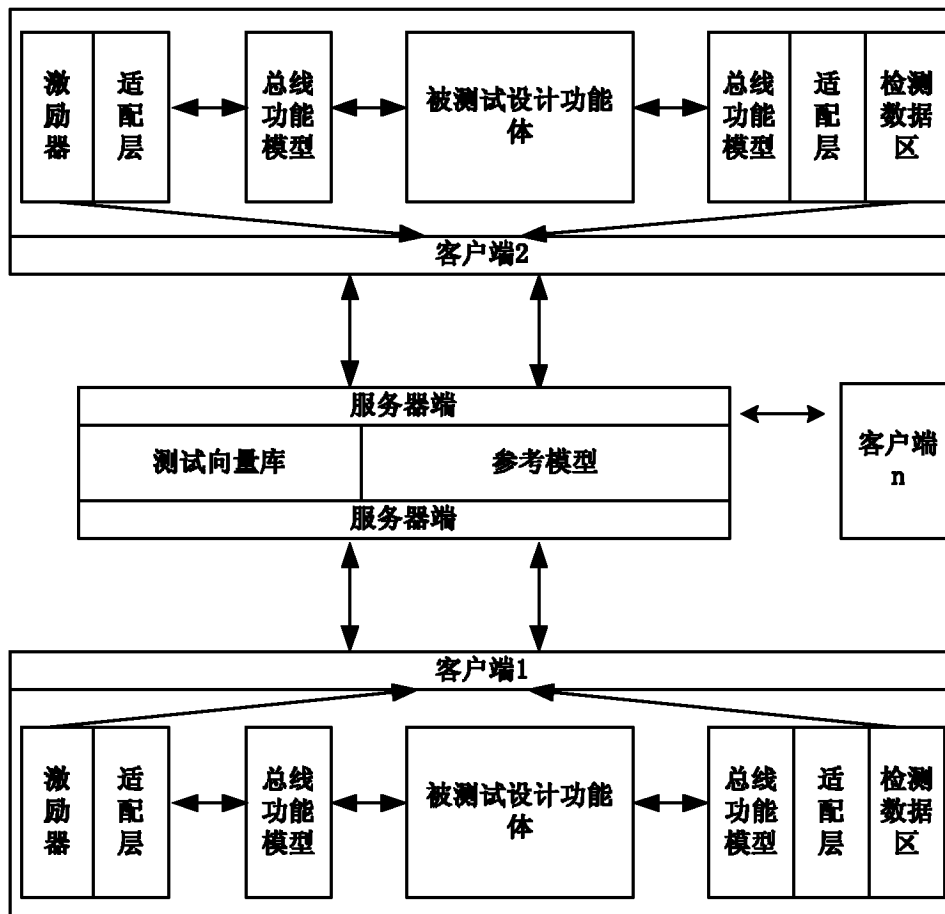


图 3

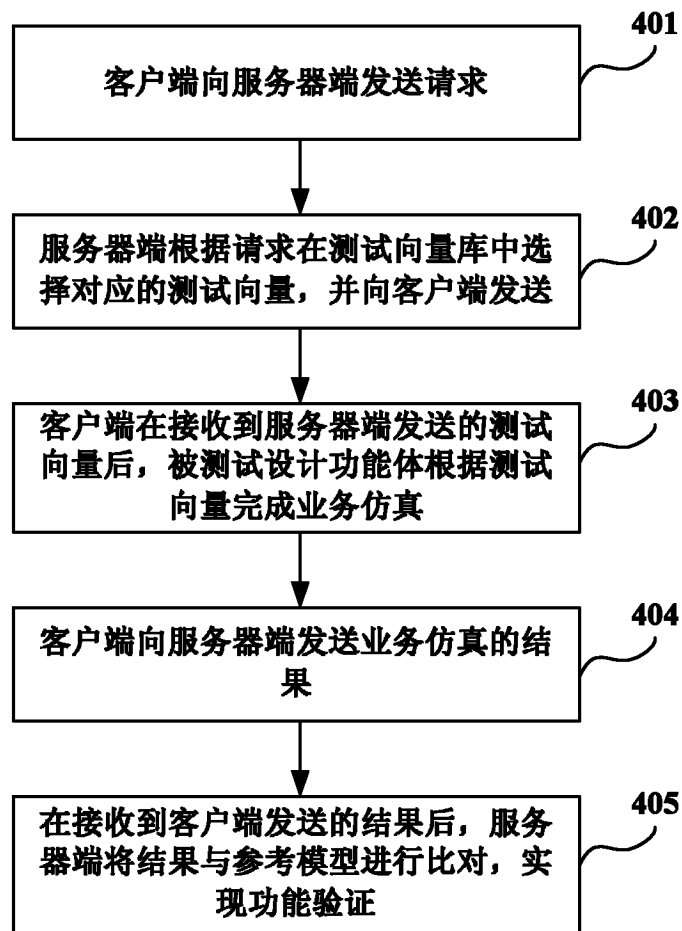


图 4