(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101133339 B
(45)授权公告日 2011.06.08

(21)申请号 200680006515.6
(22)申请日 2006.03.02
(30)优先权数据
11/070,957 2005.03.02 US
(85)PCT申请进入国家阶段日
2007.08.30
(86)PCT申请的申请数据
PCT/JP2006/304517 2006.03.02
(87)PCT申请的公布数据
WO2006/093327 EN 2006.09.08
(73)专利权人 株式会社爱德万测试
地址 日本东京
(72)发明人 安卡恩·普兰马尼克
马克·埃尔斯顿 足立敏明
(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 王允方 刘国伟
(51)Int.Cl.
G01R 31/319(2006.01)
(56)对比文件
EP 1237073 A3,2002.09.04, 全文.
US 2004/0225459 A1,2004.11.11, 摘要、附图2、说明书第0033,0064-0066,0103-0105,0108,0990-0998段.
审查员 苗文

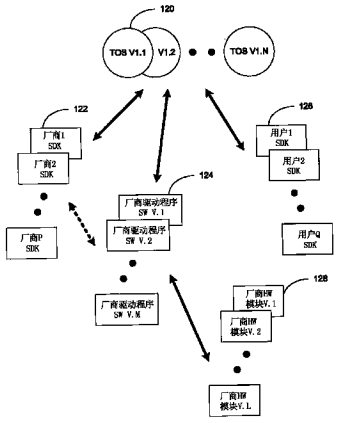
权利要求书 4 页 说明书 8 页 附图 5 页

(54)发明名称

开放式结构测试系统中厂商硬件模块修正的简洁表示

(57)摘要

本发明揭示一种用于在开放式结构测试系统中使软件组件与厂商硬件模块版本相关联的方法和系统。所述方法包含:接收厂商硬件模块的一组硬件版本;接收所述厂商硬件模块所支持的一组软件组件;处理所述组硬件版本,其中使用掩码值将所述组硬件版本表示为等效类别的硬件版本号;获得所述厂商硬件模块的硬件版本的用户选择;验证所述厂商硬件模块的硬件版本的用户选择;以及根据所述硬件版本的用户选择创建系统简档。



CN 101133339 B

1. 一种在开放式结构测试系统中用于引进新的厂商硬件模块以及其所有随附软件的方法，其包括：

接收厂商硬件模块的一组硬件版本；

接收所述厂商硬件模块所支持的一组软件组件；

处理所述组硬件版本，其中使用掩码值将所述组硬件版本表示为等效类别的硬件版本号，其中处理所述组硬件版本包括：

界定掩码值，其中所述掩码值包括遮掩号和掩码，其中所述掩码是所述组硬件版本的范围内的数据模式，且其中所述数据模式至少包括无关位和未遮掩位；以及

根据所述掩码和所述遮掩号指定等效类别的硬件版本号；

获得所述厂商硬件模块的硬件版本的用户选择；

验证所述厂商硬件模块的硬件版本的所述用户选择；以及

根据硬件版本的所述用户选择创建系统简档。

2. 根据权利要求 1 所述的方法，其中所述组软件组件包括：

一个或一个以上厂商软件开发套件；

一个或一个以上用户软件开发套件；以及

一个或一个以上厂商驱动程序软件版本，其对应于所述厂商硬件模块的所述组硬件版本。

3. 根据权利要求 1 所述的方法，其中指定所述等效类别的硬件版本号包括：

针对所述掩码中的每一无关位，遮掩所述遮掩号中的相应位，进而指示所述等效类别中的所有成员在相应的位位置具有 1 或 0；以及

针对所述掩码中的每一未遮掩位，使用所述等效类别中所述遮掩号的所述相应位，进而指示所述遮掩号中的所述相应位未被遮掩。

4. 根据权利要求 1 所述的方法，其中验证所述厂商硬件模块的硬件版本的用户选择包括：

如果所述掩码值涵盖用户选择，那么将对应于所述掩码值的整个等效类别识别为已被选择。

5. 根据权利要求 4 所述的方法，其中验证所述厂商硬件模块的硬件版本的所述用户选择进一步包括：

针对硬件版本的每一用户选择，当且仅当硬件版本的所述用户选择与所述掩码的逐位“与”等于所述遮掩号时，将硬件版本的所述用户选择表示为有效。

6. 根据权利要求 1 所述的方法，其中创建系统简档包括：

根据所述厂商硬件模块的硬件版本的所述用户选择来确定一组兼容的软件组件；

针对所述厂商硬件模块的硬件版本的每一用户选择而选择软件组件的版本；

针对所述厂商硬件模块的硬件版本的每一用户选择而创建用户选择的软件组件的列表；以及

存储所述用户选择的软件组件的列表。

7. 根据权利要求 6 所述的方法，其中选择软件组件的版本包括：

询问用户以从所述组兼容的软件组件中选择软件组件的版本；以及

针对软件组件的所述版本的每一用户选择，计算与硬件版本的所述用户选择的交集，

以产生所支持的硬件版本的列表。

8. 根据权利要求 7 所述的方法，其中计算交集包括：

使用硬件版本的所述用户选择来计算所产生的掩码；以及

使用硬件版本的所述用户选择来计算所产生的遮掩号。

9. 根据权利要求 1 所述的方法，其进一步包括：

提供开放式结构测试系统，其中所述开放式结构测试系统包括系统控制器、一个或一个以上的现场控制器、一个或一个以上的厂商硬件模块，以及一个或一个以上的待测试装置 (DUT)；

接收厂商硬件模块的新硬件版本，其中所述新硬件版本是所述等效类别的硬件版本号的成员；以及

使用所述系统简档来测试所述厂商硬件模块的所述新硬件版本。

10. 一种开放式结构测试系统，其包括：

系统控制器；

现场控制器，其耦合到所述系统控制器；

一个或一个以上的硬件模块，其耦合到所述现场控制器；

一个或一个以上的待测试装置 (DUT)，其并行耦合到硬件模块；

用于接收厂商硬件模块的一组硬件版本的构件；

用于接收所述厂商硬件模块所支持的一组软件组件的构件；

用于处理所述组硬件版本的构件，其中使用掩码值将所述组硬件版本表示为等效类别的硬件版本号，其中所述用于处理所述组硬件版本的构件包括：

用于界定掩码值的构件，其中所述掩码值包括遮掩号和掩码，其中所述掩码是所述组硬件版本的范围内的数据模式，且其中所述数据模式至少包括无关位和未遮掩位；以及

用于根据所述掩码和所述遮掩号来指定硬件版本号的等效类别的构件；

用于获得所述厂商硬件模块的硬件版本的用户选择的构件；

用于验证所述厂商硬件模块的硬件版本的所述用户选择的构件；以及

用于根据硬件版本的所述用户选择来创建系统简档的构件。

11. 根据权利要求 10 所述的系统，其中所述组软件组件包括：

一个或一个以上的厂商软件开发套件；

一个或一个以上的用户软件开发套件；以及

一个或一个以上的厂商驱动程序软件版本，其对应于所述厂商硬件模块的所述组硬件版本。

12. 根据权利要求 10 所述的系统，其中所述用于指定所述等效类别的硬件版本号的构件包括：

针对所述掩码中的每一无关位，用于遮掩所述遮掩号中的相应位进而指示所述等效类别的所有成员在相应的位位置具有 1 或 0 的构件；以及

针对所述掩码中的每一未遮掩位，用于在所述等效类别中使用所述遮掩号的所述相应位进而指示未遮掩所述遮掩号中的所述相应位的构件。

13. 根据权利要求 10 所述的系统，其中所述用于验证所述厂商硬件模块的硬件版本的所述用户选择的构件包括：

如果所述掩码值涵盖用户选择，用于将对应于所述掩码值的整个等效类别识别为已被选择的构件。

14. 根据权利要求 13 所述的系统，其中所述用于验证所述厂商硬件模块的硬件版本的所述用户选择的构件进一步包括：

针对硬件版本的每一用户选择，用于当且仅当硬件版本的所述用户选择与所述掩码的逐位“与”等于所述遮掩号时，将硬件版本的所述用户选择表示为有效的构件。

15. 根据权利要求 10 所述的系统，其中所述用于创建系统简档的构件包括：

用于根据所述厂商硬件模块的硬件版本的所述用户选择来确定一组兼容的软件组件的构件；

用于针对所述厂商硬件模块的硬件版本的每一用户选择而选择软件组件的版本的构件；

用于针对所述厂商硬件模块的硬件版本的每一用户选择而创建用户选择的软件组件的列表的构件；以及

用于存储所述用户选择的软件组件的列表的构件。

16. 根据权利要求 15 所述的系统，其中所述用于选择软件组件的版本的构件包括：

用于询问用户以从所述组兼容的软件组件中选择软件组件的版本的构件；以及

针对软件组件的所述版本的每一用户选择，用于计算与硬件版本的所述用户选择的交集以产生所支持的硬件版本的列表的构件。

17. 根据权利要求 16 所述的系统，其中所述用于计算交集的构件包括：

用于使用硬件版本的所述用户选择来计算所产生的掩码的构件；以及

用于使用硬件版本的所述用户选择来计算所产生的遮掩号的构件。

18. 根据权利要求 10 所述的系统，其进一步包括：

用于提供开放式结构测试系统的构件，其中所述开放式结构测试系统包括系统控制器、一个或一个以上的现场控制器、一个或一个以上的厂商硬件模块，以及一个或一个以上的待测试装置 (DUT)；

用于接收厂商硬件模块的新硬件版本的构件，其中所述新硬件版本是所述等效类别的硬件版本号的成员；以及

用于使用所述系统简档来测试所述厂商硬件模块的所述新硬件版本的构件。

19. 一种在开放式结构测试系统中用于引进新的厂商硬件模块以及其所有随附软件的方法，其中所述开放式结构测试系统包括系统控制器、现场控制器、一个或一个以上的厂商硬件模块，以及一个或一个以上的待测试装置 (DUT)，所述方法包括：

接收厂商硬件模块的一组硬件版本；

使用包括遮掩号和掩码的掩码值将所述组硬件版本表示为硬件版本号的等效类别；以及

其中所述掩码是所述组硬件版本的范围内的数据模式；

其中所述数据模式至少包括无关位和未遮掩位；以及

其中所述将所述组硬件版本表示为硬件版本号的等效类别进一步包括：

针对所述掩码中的每一无关位，遮掩所述遮掩号中的相应位，进而指示所述等效类别的所有成员在相应的位位置具有 1 或 0；以及

针对所述掩码中的每一未遮掩位，使用所述等效类别中所述遮掩号的所述相应位，进而指示所述遮掩号中的所述相应位未被遮掩。

20. 根据权利要求 19 所述的方法，其进一步包括：

获得所述厂商硬件模块的硬件版本的用户选择；以及

验证所述厂商硬件模块的硬件版本的所述用户选择。

21. 根据权利要求 20 所述的方法，其中验证所述厂商硬件模块的硬件版本的所述用户选择进一步包括：

针对硬件版本的每一用户选择，当且仅当硬件版本的所述用户选择与所述掩码的逐位“与”等于所述遮掩号时，将硬件版本的所述用户选择表示为有效；以及

如果用户选择确定为有效，那么将对应于所述掩码值的整个等效类别识别为已被选择。

22. 根据权利要求 19 所述的方法，其进一步包括：

接收所述厂商硬件模块的新硬件版本；以及

使用所述等效类别的硬件版本号来测试所述厂商硬件模块的所述新硬件版本是否是所述等效类别的硬件版本号的成员。

开放式结构测试系统中厂商硬件模块修正的简洁表示

技术领域

[0001] 本发明涉及用于半导体测试的自动化测试设备 (ATE) 的领域。明确地说，本发明涉及简洁表示开放式结构测试系统中的厂商硬件模块版本以实现有效的字段部署。

背景技术

[0002] 在开放式结构测试系统中，对于用户的特定系统所配置有的每个厂商硬件模块并且对于每一此类厂商硬件模块的每个不同的相关联的模块软件组件，软件需要访问保证与相应的模块软件的每一版本一起工作的厂商硬件模块版本的列表的厂商规格。因而，系统需要有效地分析此复杂的信息，以将其应呈现给用户的对软件组件的选择确定为系统软件内容的候选者。

[0003] 理想的情况是，所呈现的对软件组件的选择应包括尽可能大的一组软件组件，全部约束条件是：保证用户从给定选择中挑选的任何完整组的软件组件一起工作，并与用户所选的硬件配置一起工作。在此问题的情况下，一种方法是为每一软件组件列举兼容的硬件版本（例如，逗号分隔的整数列表），以表示硬件版本号（其中范围分类符用于表示连续的号）。此方法存在许多问题。

[0004] 首先，硬件版本号通常较大；需要多达 32 位，其中所述号内的不同位字段对于硬件厂商来说可能具有不同的所有权意义。如果厂商选择针对号内与单个软件版本兼容的位的特定模式用所有可能值认证所有版本，那么这可能导致：a) 版本号的非连续序列，其无法通过范围分类符来有效地表示；和 / 或 b)：数目非常大的版本号，从而产生非常长的列表，其对于存储和 / 或处理目的而言可能相当不切实际。第二，所述方法尤其在非常频繁的硬件产品更新的情况下相当具有局限性，所述硬件产品更新可能要求对列表进行频繁添加，且因此，要求对用户处系统简档进行冗长且频繁的再生过程来合并所述添加。因此，需要简洁表示开放式结构测试系统中的厂商硬件模块版本以实现有效的字段部署。

发明内容

[0005] 揭示一种方法：在开放式结构测试系统中，硬件模块厂商可以简洁的方式指定保证与相关联的模块软件的给定版本一起工作的硬件版本号的任意序列。另外，对硬件版本号进行处理以便为用户提供针对厂商硬件模块的一组最佳的软件组件选择，其产生用于字段中的部署的有效系统配置。

[0006] 在一个实施例中，一种在开放式结构测试系统中使软件组件与厂商硬件模块版本相关联的方法包含：接收厂商硬件模块的一组硬件版本；接收所述厂商硬件模块所支持的一组软件组件；处理所述组硬件版本，其中使用掩码值将所述组硬件版本表示为等效类别的硬件版本号；获得所述厂商硬件模块的硬件版本的用户选择；验证所述厂商硬件模块的硬件版本的用户选择；以及根据所述硬件版本的用户选择创建系统简档。

[0007] 在另一实施例中，一种开放式结构测试系统包含系统控制器、耦合到所述系统控制器的现场控制器、耦合到所述现场控制器的一个或一个以上硬件模块，以及由多组硬件

模块并行服务的一个或一个以上待测试装置 (DUT)。所述系统进一步包含 :用于接收厂商硬件模块的一组硬件版本的构件 ;用于接收所述厂商硬件模块所支持的一组软件组件的构件 ;用于处理所述组硬件版本的构件,其中使用掩码值将所述组硬件版本表示为等效类别的硬件版本号 ;用于获得所述厂商硬件模块的硬件版本的用户选择的构件 ;用于验证所述厂商硬件模块的硬件版本的用户选择的构件 ;以及用于根据所述硬件版本的用户选择创建系统简档的构件。

[0008] 在又一实施例中,一种在开放式结构测试系统中表示厂商硬件模块版本的方法包含 :接收厂商硬件模块的一组硬件版本 ;以及使用包含掩码号和掩码的掩码值将所述组硬件版本表示为等效类别的硬件版本号。所述掩码是所述组硬件版本的范围内的数据模式,且所述数据模式至少包含无关位 (don′ t-care bit) 和未遮掩位 (unmask bit)。

附图说明

[0009] 结合以下图式在阅读对本发明实施例的详细描述之后将更清楚地可理解本发明的上述特征和优点以及其额外的特征和优点。

[0010] 图 1a 说明根据本发明实施例用于以一个或一个以上 SiteC 并行测试多个 DUT 的开放式结构测试系统。

[0011] 图 1b 说明开放式结构测试系统中多个厂商硬件模块版本及其相应的软件组件版本的实例。

[0012] 图 2 说明根据本发明实施例将掩码应用于遮掩号。

[0013] 图 3 说明根据本发明实施例创建用于在开放式结构测试系统中测试不同的厂商硬件模块版本的系统简档的方法。

[0014] 图 4 说明根据本发明实施例计算被遮掩表示的交集的方法。

具体实施方式

[0015] 本发明提供用于简洁表示开放式结构测试系统中的厂商硬件模块版本的方法和系统。提供以下描述以使所属领域的技术人员能够制造和使用本发明。特定实施例和应用的描述仅作为实例而提供。所属领域的技术人员将容易了解本文描述的实例的各种修改和组合,且本文定义的一般原理可应用于其它实例和应用。因此,不希望本发明限于所描述和展示的实例,而是本发明应符合与本文揭示的原理和特征一致的最广泛范围。

[0016] 一种可再配置的开放式结构测试系统提供可开发、个别认证第三方软件和硬件(也称为厂商硬件模块)并将其可靠地整合到开放式结构测试系统中的解决方案。图 1a 说明根据本发明实施例用于以一个或一个以上 SiteC 并行测试多个 DUT 的开放式结构测试系统。所述系统包含系统控制器 102、多个现场控制器 104、多个厂商硬件模块 106,以及相应组的 DUT 110 ;每一组 DUT 可包含一个或一个以上个别 DUT 112。测试头 108 固持所述组 DUT 110。在一个实施例中,系统控制器和现场控制器可由具有处于多个地理位置的多个计算机的分布式计算系统来实施。在另一实施例中,系统控制器和现场控制器可由单个计算机来实施。第 60/447,839 号美国申请案“Method and Structure to Develop aTest Program for Semiconductor Integrated Circuits”和第 60/573,577 号美国申请案“Software Development in an Open Architecture Test System”提供了对开放式结构测试系统的详

细描述。另外，A.Pramanick 等人在 2004 年 10 月的 2004IEEE 国际测试会议中的“Testing Programming Environment in a Modular Open Architecture Test System”第 413-422 页也描述了开放式结构测试系统。

[0017] 在这种开放式结构测试系统中，每一模块化单元（软件与硬件）可由第三方单元代替，只要第三方解决方案符合整合框架的要求。因此，第三方硬件可以是任何功能单元，例如数字引线卡、模拟卡、装置电源等。术语厂商硬件模块（或简称为模块）用于指此类硬件功能单元，而支持厂商硬件模块的相关联的厂商软件组件被称作模块软件。后者包括若干不同的软件组件，例如模块控制驱动程序软件、校准和诊断软件、仿真软件、厂商特定模式编译器等。

[0018] 开放式结构测试系统不仅用于测试系统的操作，而且用于控制测试环境并引入新的厂商硬件模块以及其所有附随软件。图 1b 说明开放式结构测试系统中多个厂商硬件模块版本 128 及其相应的软件组件版本的实例。软件组件包含测试装置操作系统 (TOS)120 的多个版本及其相关联的厂商软件开发套件 (SDK)122 和用户 SDK126 的版本，以及厂商软件驱动程序 124 的多个版本。第 60/635,094 号美国申请案“Method and System forPerforming Installation and Configuration Management of Tester Instrument Modules”提供了管理开放式结构测试系统中例如 TOS、厂商 SDK 和用户 SDK 的多个厂商硬件模块版本和软件组件版本的方法，所述申请案转让给本申请案的受让人。

[0019] 测试系统需要足够灵活以使得针对厂商硬件模块 M 的模块软件的特定版本 n 可由与特定系统配置兼容的测试系统来检验，只要模块 M 的厂商已认证软件的版本 n 与可能在所述给定系统配置中共存的厂商硬件模块的某些不同版本兼容即可。厂商需要公布保证与针对 M 的模块软件的版本 n 一起工作的模块 M 的硬件版本的列表。

[0020] 当用户以厂商硬件模块及其相关联的模块软件的特定版本来配置开放式结构测试系统时，由系统软件来检验用户的选择实际上已形成工作系统。由于这个原因，此类开放式结构测试系统需要依赖于由用户创建的系统简档信息，在简档创建期间由系统给用户提供正确的引导，所述引导约束用户仅对系统配置（硬件和软件两者）进行有效选择，且假定大型系统可具有数十种不同厂商的硬件模块，每一厂商硬件模块均具有多个相关联的软件组件。因此，向用户提供正确引导以创建系统简档同时不会不必要地限制用户的任务非常复杂。

[0021] 大体上，提供开放式结构测试系统中的厂商硬件模块版本的简洁表示以实现有效的字段部署的方法包含两个主要部分 :1) 指定硬件版本 ;和 2) 处理硬件版本信息并允许用户创建有效的系统简档。所属领域的技术人员将了解，可应用所述方法来表示开放式结构测试系统中的软件组件版本，例如 TOS、厂商 SDK 和用户 SDK 的版本。

[0022] 在一个实施例中，通过遮掩硬件版本号中任意组（厂商指定）的位来实现兼容的硬件版本规格。举例来说，硬件版本号的语法可在规格文件中表示如下 :1095188784, 0x01403500/0x0ff0ff00,8800000111，其中号码“1095188784”和“8800000111”表示未遮掩的单个硬件版本，而号码“0x01403500/0x0ff0ff00”表示遮掩号，其表示整个系列（类别）的等效硬件版本号。

[0023] 在此实例中，以十六进制格式、语法“n/m”来表示遮掩值，其中“n”表示遮掩号，且“m”表示掩码。表达式“n/m”中遮掩号 n 的掩码 m 被定义为位模式，其指定基于 n 的等效

类别 (EQC) 的号，表示为“EQC(n/m)”，使得 1) 对于 m 中无关的每一位（此实例中由“0”位表示），n 中相应的位被遮掩，从而指示等效类别的所有成员可在所述位位置具有 1 或 0；和 2) 对于 m 中作为未遮掩位的每一位（此实例中由“1”位表示），等效类别的任何号码中相应的位与 n 中相应的位相同。

[0024] 图 2 说明根据本发明实施例将掩码应用于遮掩号。考虑图 2 中的元组“010/110”（为便于说明，以二进制格式而不是十六进制格式表达）。将掩码 m202 应用于号码 n204 以创建等效类别的版本号 206。

[0025] 应用以上给定的掩码的定义，号码 x(x 为 EQC(010/110) 的成员）在位位置 2 和 3 中分别具有“1”和“0”（从右（或最低有效）向左（或最高有效）数三个位）。这是因为，掩码“110”中位位置 2 和 3 为 1（即，未遮掩），其暗示在 x 中，这两个位具有 n 中相应位的准确值。然而，x 在位位置 1 可具有“1”或“0”，因为掩码“110”中位 1 的值为“0”（即，被遮掩）。因此，EQC(010/110) 是集合 {010,011}，即集合 {2,3} 以及因此 x 可以是 2 或 3。

[0026] 在开放式结构测试系统中，通常执行的操作是确定用户所提供的硬件版本号 x 是否为被遮掩的等效类别的成员。确定被遮掩的等效类别中的成员资格的一种方式是，展示当且仅当 x & m = n（其中“&”是逐位“与”运算符）时号码 x 是 EQC(n/m) 的成员。此特性提供执行所需确定的便利的方式。

[0027] 接下来，描述以大量非连续版本号的等效类别的遮掩表示来产生系统简档的方法。大体上，所述方法包含：1) 处理所述组被遮掩硬件版本；2) 向用户提供兼容软件组件版本的选择以运行厂商硬件模块的用户选择的版本；和 3) 计算由于用户选择模块的兼容软件组件版本而产生的每一厂商硬件模块的所述组硬件版本的简洁表示；和 4) 将简洁表示存储在用户的系统简档中。

[0028] 对于处理所述组被遮掩硬件版本，如下描述用于表示厂商硬件模块及其相关联的软件组件的版本的符号。首先将特定厂商硬件模块表示为 M，且由集合 {MX} 来表示需要用于控制 ATE 中的硬件模块 M 的软件组件，其中 X 可以是 A、B、C、... 等。因此，举例来说，如果 M 总共具有四个软件组件，那么将它们表示为 MA、MB、MC 和 MD。接下来，将厂商硬件模块 M 的软件组件 X 的特定版本 k 表示为 MX.k。举例来说，将模块 M 的组件 A 的版本 1 表示为 MA.1。此外，将厂商硬件模块 M 的软件组件 X 的第 k 个版本所支持的该组硬件版本表示为 HWR(MX.k)。举例来说，HWR(MA.1) 表示集合 {3,5,010/011}。注意，为了简便，以二进制格式表示遮掩号。所属领域的一般技术人员将了解，可以十六进制或其它格式表示遮掩号。

[0029] 图 3 说明根据本发明实施例创建用于在开放式结构测试系统中测试不同的厂商硬件模块版本的方法。所述方法在开始状态开始 (302) 并随后移动到步骤 1(304)，其中所述方法询问用户对于测试系统中模块 M 的硬件版本号的选择。如上所述，将任意遮掩的值用于硬件版本和所产生的硬件版本号系列会在向用户表示等效类别 EQC(n/m) 时形成挑战。所有可能值 x(x 是 EQC(n/m) 的成员）的完全枚举（和随后的维护）可呈指数形式的复杂，从 m 中指定的遮掩位的数目来看尤其如此。甚至对于 m 的适度较大的值（例如，对于 m 小于或等于 32)，完全枚举的方法显然无效且不切实际。因此，提示用户输入其对硬件版本值的选择。

[0030] 在步骤 2(306) 中，所述方法确定每一用户输入是否为有效选择。使用上文描述的用于确定硬件版本号是否为 EQC 的成员的方法来执行所述确定。如果用户输入有效 (306_

是），那么所述方法继续到步骤 3。如果无效 (306_ 否），那么所述方法重复步骤 1 并询问用户再输入硬件版本号。

[0031] 在步骤 3(308) 中，当用户成功输入硬件版本号时，所述方法确定与 HWR(MX.k)（用户对 M 的硬件版本的选择）兼容的 X 的软件组件 k 的一组兼容版本。在每一步骤中，使用上文论述的用于针对遮掩值来测试硬件版本号的方法。对于每一用户选择；如果其由遮掩值涵盖，那么认为对应于所述遮掩值的整个等效类别已被选择。此特性对于实现硬件版本号的有效简洁表示很重要。

[0032] 在步骤 4(310) 中，所述方法询问用户选择来自以上确定的所述组兼容软件组件的软件组件的一个版本。

[0033] 在步骤 5(312) 中，所述方法计算 HWR(MX.p) 与所有 X 的交集，其中 p 表示针对来自步骤 1 的每一硬件版本选择的来自 X 的软件组件的所选择的版本。下文结合对图 4 的论述给出交集的计算的详细论述。此步骤通过针对软件组件 X 的每一用户选择的版本 p 来计算所有集合 HWR(MX.p) 的交集来估计所有兼容硬件版本的整个列表。此步骤针对用户的硬件版本选择的每一者重复此计算（其中每一用户选择产生单个值或单个 EQC)，并收集结果。

[0034] 在步骤 6(314) 中，所述方法使用来自步骤 5 的对所有选择的硬件版本的结果集合来计算与用户的软件版本选择兼容的 M 的硬件版本的最终列表。

[0035] 在步骤 7(316) 中，所述方法将以上计算出的最终列表存储在用户的简档中。所述方法在结束状态中结束 (318)。

[0036] 图 4 说明根据本发明实施例计算遮掩表示的交集的方法。如图 4 所示，计算两个组件 m_I 与 m_J 的交集（假定每一选择均是被遮掩的选择）（即，整个等效类别）。所属领域的一般技术人员将了解，可能将用于计算交集的方法延伸到硬件版本号的未遮掩与遮掩或未遮掩与未遮掩选择。如图 4 中所示，用户对 m_I 的选择由 $EQC(n_I/m_I)$ 表示，且用户对 m_J 的选择由 $EQC(n_J/m_J)$ 表示。使交集的结果为 EQC(n/m)。

[0037] 所述方法在开始状态开始 (402) 并随后移动到步骤 1(404)，其中所述方法计算所产生的掩码。符号“1”表示逐位“或”运算。m_I 与 m_J 的逐位“或”的结果正是所产生的掩码，因为按照逐位“或”运算的定义和掩码的定义，其是对 m_I 与 m_J 两者均满足的唯一的掩码。应注意，如果 m 为空（即，全为 1)，那么 EQC(n/m) 由单个值 n 组成。在步骤 2(406) 中，所述方法计算所产生的遮掩号 n。所述方法在结束状态结束 (408)。所属领域的一般技术人员将了解，为了将以上交集的计算延伸到两个以上组件，可使用步骤 2 中计算出的 n 和 m 的值作为用于计算中下一迭代的初始值，等等。

[0038] 以下实例应用上述图 3 和 4 中描述的方法。假定特定厂商硬件模块 M 具有三个软件组件：组件 MA、组件 MB 和组件 MC。还假定表 1 中给出了特定用户对于这些组件的可用软件版本及其所支持的相应硬件版本的一组选择，其中符号“MX.n”表示软件组件 MX 的第 n 个版本。为便于说明，以二进制而不是十六进制表示遮掩号。另外，假定所述组所有可能的硬件版本可由三个位表示。

[0039]

模块 M 的 SW 组件	SW 组件所支持的硬件版本
MA.1	3,5,010/011
MA.2	5,000/100
MA.3	000/110
MB.1	010/010,5
MB.2	000/000
MC.1	5,010/110

[0040] 表 1

[0041] 应注意，最终目标是允许用户正确地选择可用于与模块 M 的硬件版本的选择一起工作的每一软件组件的一个版本（即，MA、MB 和 MC 各一个版本）。

[0042] 将图 3 的方法应用于以上实例：

[0043] 1. 针对整个集合 MX.n，即 {3,5,010/011,000/100,000/110,010/010,000/000,010/110}，测试用户选择。

[0044] 假设用户选择 M 的硬件版本“2”和“5”。这导致以下关联：

[0045] a. 对于硬件版本“2”，关联是（即，集合 $HWR(MX.n)_2$ 是）：

[0046]

$$MA.1 \Leftrightarrow 010/011$$

[0047]

$$MA.2 \Leftrightarrow 000/100$$

[0048]

$$MA.3 \Leftrightarrow \emptyset$$

[0049] (MA.3 不支持“2”)

[0050]

$$MB.1 \Leftrightarrow 010/010$$

[0051]

$$MB.2 \Leftrightarrow 000/000$$

[0052]

$$MC.1 \Leftrightarrow 010/110$$

[0053] b. 对于硬件版本“5”，关联是（即，集合 $HWR(MX.n)_5$ 是）：

[0054]

$$MA.1 \Leftrightarrow 5$$

[0055]

$$MA.2 \Leftrightarrow 5$$

[0056] $MA.3 \Leftrightarrow \emptyset$ (MA.3 不支持“5”)

[0057]

$$MB.1 \Leftrightarrow 5$$

[0058]

$$MB.2 \Leftrightarrow 000/000$$

[0059]

$$MC.1 \Leftrightarrow 5$$

[0060] 2. 以上关联指示用户的可用软件组件选择是

[0061] {MA.1,MA.2} 之一,

[0062] {MB.1,MB.2} 之一,以及

[0063] {MC.1} 之一。

[0064] 3.假设用户(从以上2中的每一组件群组中选择一个)选择以下组件:MA.2、MB.2和MC.1。那么,

[0065] a.对于硬件版本选择“2”,根据以下步骤由集合 $HWR(MA.2)_2$、$HWR(MB.2)_2$ 和 $HWR(MC.1)_2$ 的交集给出所支持的硬件版本的列表:

[0066] i.使用如下图4的方法计算 $HWR(MA.2)_2$ 与 $HWR(MB.2)_2$ 的交集:$000/100 \cap 000/000 \Rightarrow 000/100$。(其中 000 & 100 = 000,且 000 & 000 = 000。)

[0067] ii.计算 000/100 与 $HWR(MC.1)_2$ 的交集:$000/100 \cap 010/110 \Rightarrow 010/110$。(其中 010&100 = 000,且 010 & 110 = 010。)

[0068] iii.因此,硬件版本“2”的选择的最终等效类别是 010/110,即 {2,3}。

[0069] b.对于硬件版本选择“5”,由集合 $HWR(MA.2)_5$、$HWR(MB.2)_5$ 和 $HW(MC.1)_5$ 的交集给出所支持的硬件版本的列表:

[0070] i.首先,使用如下图3的方法计算 $HWR(MA.2)_5$ 与 $HWR(MB.2)_5$ 的交集:$5 \cap 000/000 \Rightarrow 101/111 \cap 000/000 \Rightarrow 101/111$。

[0071] ii.接着,计算 101/111 与 $HWR(MC.1)_5$ 的交集:$101/111 \cap 5 \Rightarrow 101/111 \cap 101/111 \Rightarrow 101/111 \Rightarrow 5$。

[0072] iii.因此,硬件版本“5”的选择的最终值是5。

[0073] 4.通过以上3a和3b,应用图3中的步骤6,因此,待包含在用户的简档中用于其对软件组件MA.2、MB.2和MC.1的选择的所支持的硬件版本的列表是 {010/110,5},即 {2,3,5}。

[0074] 因此,在开放式结构测试系统中,硬件模块厂商可指定保证与相关联的模块软件的给定版本一起工作的硬件版本号的任意序列。另外,对硬件版本号进行处理以便为用户提供针对厂商硬件模块的一组最佳的软件组件选择,其产生用于字段中的部署的有效的系统配置。

[0075] 本发明至少提供以下益处。首先,使用所支持的模块硬件版本的简洁表示,从而表示厂商硬件模块的实际所支持的产品版本的整个系列。第二,所揭示的方法允许厂商公布兼容的软件组件的硬件版本规格,其同时支持由指定值隐含指示的所有硬件版本。第三,厂商可指定任意位模式,使得厂商不限于仅选择连续的版本号。第四,厂商指定的值可由开放式结构测试系统视为组成所述组的所有个别硬件版本号的简洁集合。第五,所揭示的方法有效地分析针对与用户选择的厂商硬件模块相关联的许多软件模块的多个兼容硬件版本列表,并向用户呈现供选择的一组最佳的兼容软件组件,同时保证正确的系统操作。最后,所揭示的方法计算每一选择的模块的整个系列的等效硬件版本的最佳简洁表示,并将其存储在用户的系统简档中。因此,用来自同一系列的另一厂商硬件模块代替一厂商硬件模块既不需要重新发布硬件版本列表也不需要重新产生系统简档。

[0076] 将了解,以上描述内容为了清楚起见已参考不同的功能单元和处理器来描述了本发明实施例。然而,将了解,可在不偏离本发明的情况下使用不同功能单元或处理器之间的功能性的任何合适分布。举例来说,说明为由单独处理器或控制器执行的功能性可由同一

处理器或控制器执行。因此，对特定功能单元的参考仅应视为对用于提供所描述的功能性的合适构件的参考，而不指示严格的逻辑或物理结构或组织。

[0077] 可以任何合适的形式实施本发明，所述形式包含硬件、软件、固件或这些的任何组合。本发明视需要可部分实施为在一个或一个以上数据处理器和/或数字信号处理器上运行的计算机软件。可在物理上、功能上和逻辑上以任何合适方式实施本发明实施例的元件和组件。事实上，所述功能性可实施在单个单元中、多个单元中，或实施作为其它功能单元的一部分。因此，本发明可实施在单个单元中，或可在物理上和功能上分布在不同的单元和处理器之间。

[0078] 相关领域的技术人员将了解，可使用所揭示的实施例的许多可能的修改和组合，同时仍使用相同的基本基础机制和方法。已出于阐释的目的参考特定实施例编写了以上描述。然而，不希望以上说明性论述为详尽的或将本发明限于所揭示的精确形式。鉴于以上教示，可能作出许多修改和变化。选择并描述所述实施例是为了阐释本发明的原理及其实践应用，且使所属领域的其他技术人员能够最佳地利用本发明和具有适于所预期的特定用途的各种修改的各种实施例。

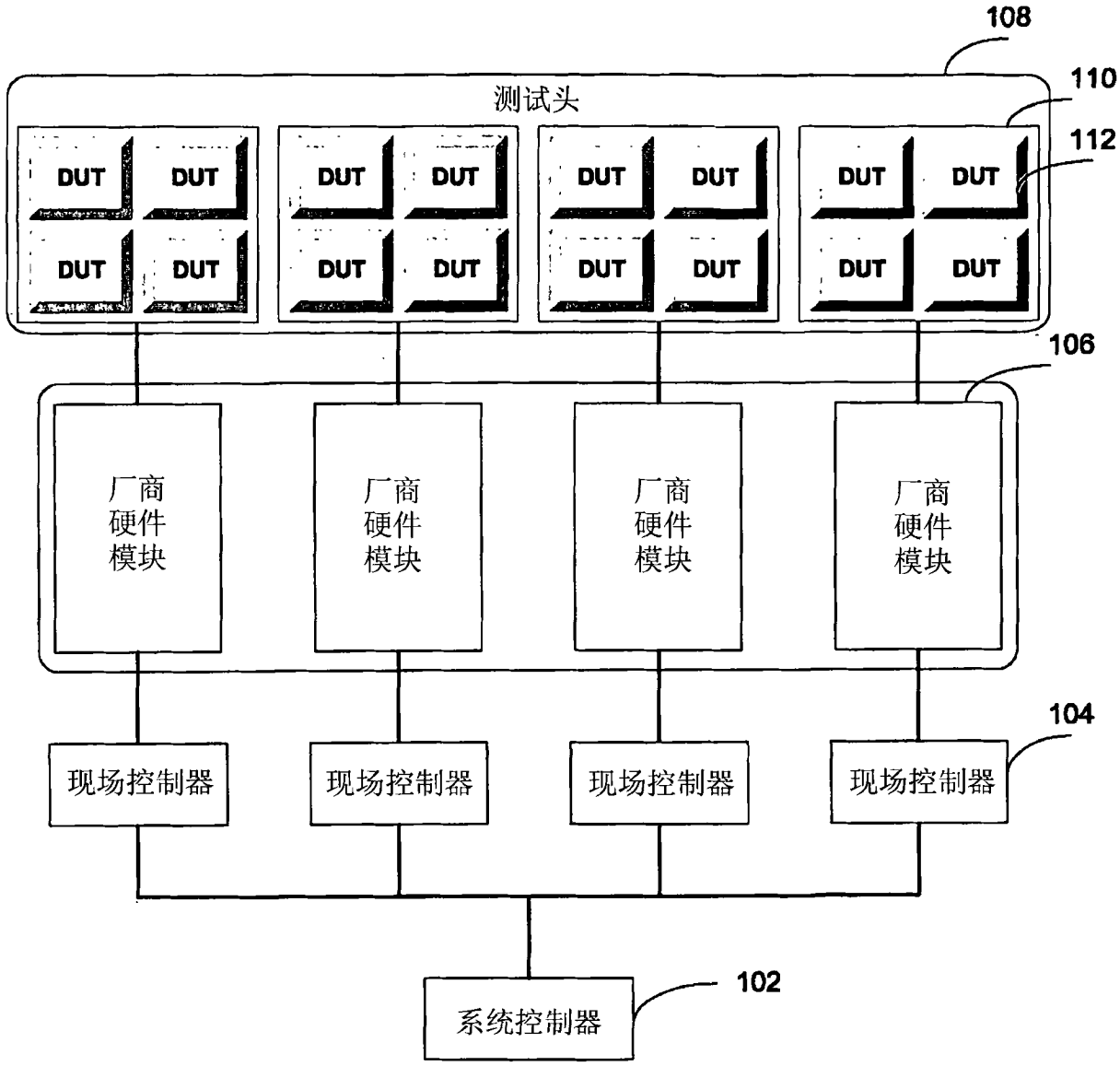


图 1a

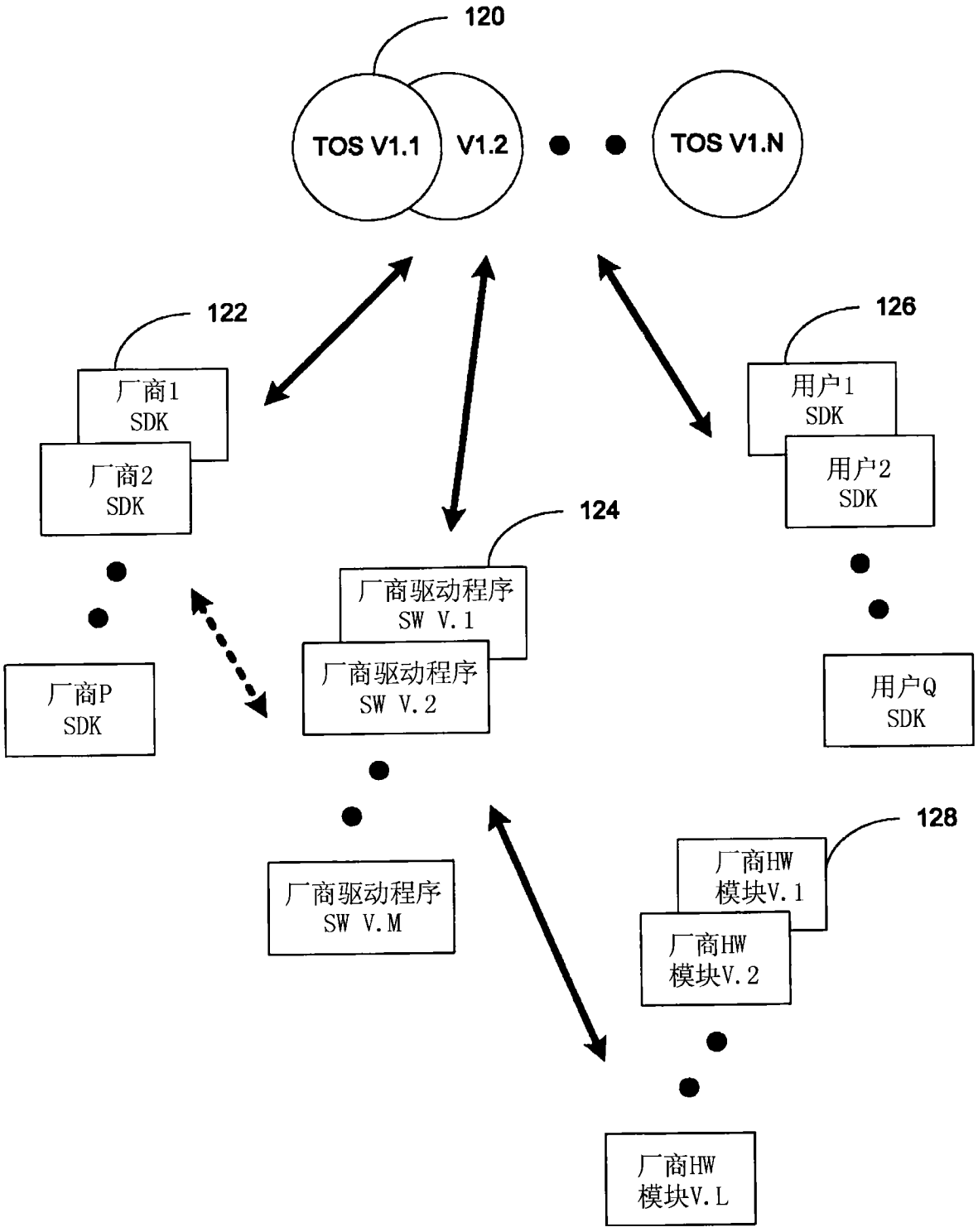

120
TOS V1.1
V1.2
TOS V1.N
122
厂商1 SDK
厂商2 SDK
厂商P SDK
124
厂商驱动程序 SW V.1
厂商驱动程序 SW V.2
厂商驱动程序 SW V.M
126
用户1 SDK
用户2 SDK
用户Q SDK
128
厂商HW 模块V.1
厂商HW 模块V.2
厂商HW 模块V.L

图 1b

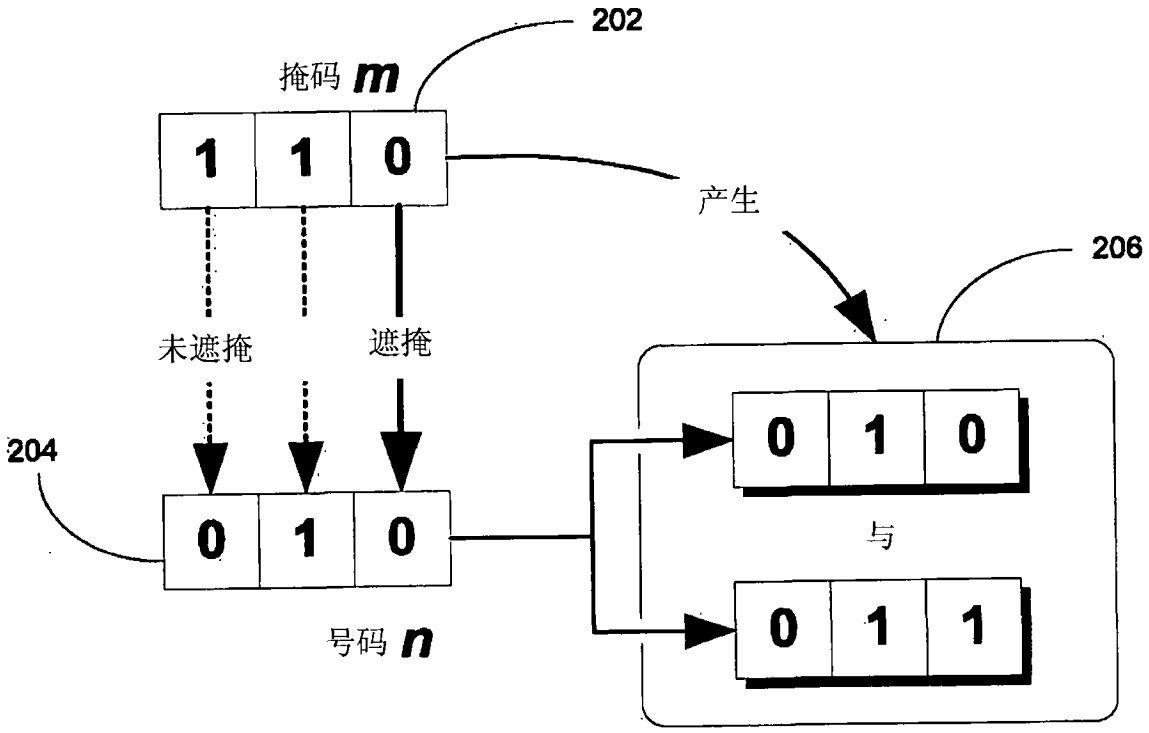


图 2

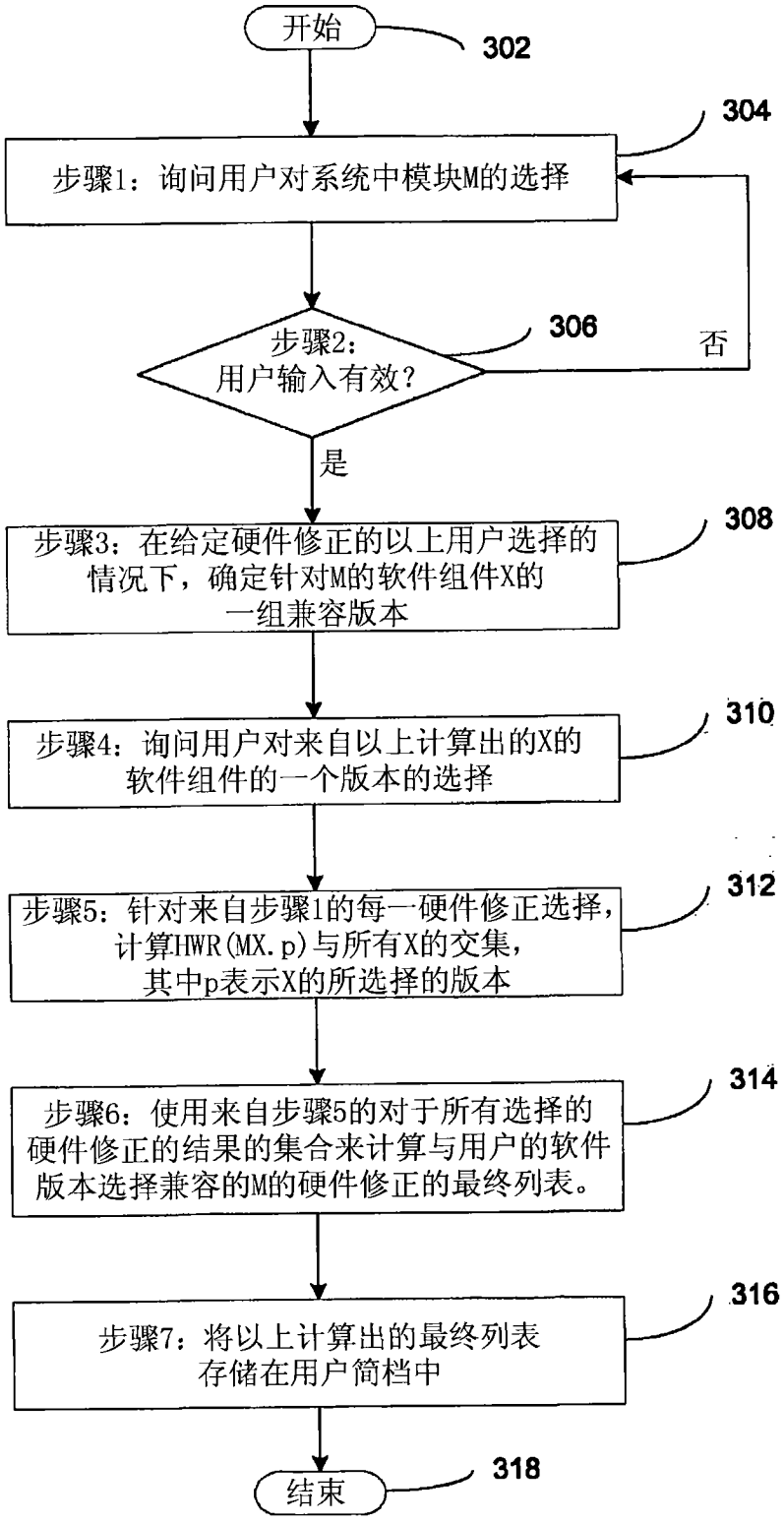

开始
302
304
步骤1：询问用户对系统中模块M的选择
步骤2：
用户输入有效？
306
否
是
步骤3：在给定硬件修正的以上用户选择的
情况下，确定针对M的软件组件X的
一组兼容版本
308
步骤4：询问用户对来自以上计算出的X的
软件组件的一个版本的选择
310
步骤5：针对来自步骤1的每一硬件修正选择，
计算HWR(MX.p)与所有X的交集，
其中p表示X的所选择的版本
312
步骤6：使用来自步骤5的对于所有选择的
硬件修正的结果的集合来计算与用户的软件
版本选择兼容的M的硬件修正的最终列表。
314
步骤7：将以上计算出的最终列表
存储在用户简档中
316
结束
318

图 3

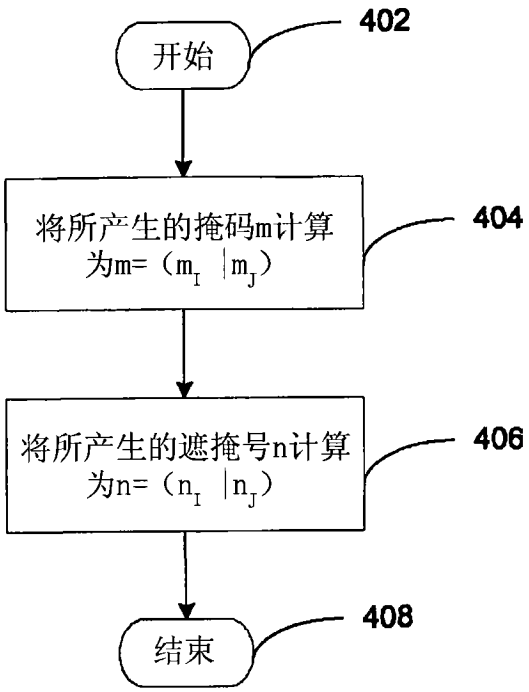


图 4