(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 101034402 B
(45)授权公告日 2010.10.27

(21)申请号 200710003107.1

(22)申请日 2007.01.31

(30)优先权数据
11/345,043 2006.01.31 US

(73)专利权人 韦瑞吉(新加坡)私人有限公司
地址 新加坡新加坡市

(72)发明人 罗伯特·斯坦利·科洛曼
瑞德·哈郝

(74)专利代理机构 北京东方亿思知识产权代理有限责任公司 11258
代理人 王怡

(51)Int.Cl.
G06F 17/30(2006.01)
G06F 11/22(2006.01)
G01R 31/317(2006.01)
G01R 31/319(2006.01)

(56)对比文件
US 2002/0078216 A1,2002.06.20, 全文.
US 5818603 A,1998.10.06, 全文.
US 6330628 B1,2001.12.11, 全文.
CN 1274890 A,2000.11.29, 全文.

审查员 庄锦军

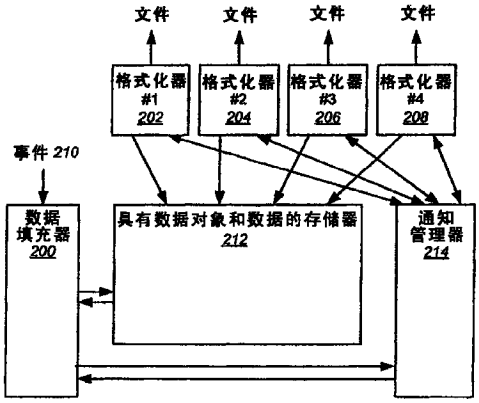
权利要求书 1 页 说明书 6 页 附图 5 页

(54)发明名称

对器件测试中生成的自定义事件进行处理的方法和装置

(57)摘要

本发明公开了用于处理在器件测试期间生成的用户自定义事件的方法和装置。响应于与对至少一个被测器件的多个测试的执行相对应的事件有序序列,1)多个数据对象被创建,数据对象包括A)与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象,以及B)与所述事件中的一个用户自定义事件相对应的通用数据对象;2)在分层树结构中使数据对象中的一些与数据对象中的其他相关,同时基于用户自定义事件在事件有序序列中的位置使通用数据对象与所述数据对象中的其他相关;以及3)在分层树结构中使与事件中的一些事件相对应的数据与数据对象中的一些相关。使多个数据格式化器可以访问与分层树结构相关联的多个数据对象和数据。



CN 101034402 B

1. 一种对器件测试中生成的自定义事件进行处理的装置，包括：

数据填充器，用来 i) 接收与对至少一个被测器件的多个测试的执行相对应的事件有序序列，ii) 创建多个数据对象，包括与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及与所述事件中的一个用户自定义事件相对应的通用数据对象，iii) 在分层树结构中使所述数据对象中的一些数据对象与所述数据对象中的其他数据对象相关，同时基于所述用户自定义事件在所述事件有序序列中的位置使所述通用数据对象与所述数据对象中的其他数据对象相关，以及 iv) 在所述分层树结构中使与所述事件中的一些事件相对应的数据与所述的数据对象中的一些数据对象相关；以及

数据格式化器，用来取回和格式化与所述分层树结构相关联的数据，所述数据格式化器还取回与所述通用数据对象相关联的数据。

2. 根据权利要求 1 所述的装置，其中，所述数据格式化器按照通用记录类型来格式化与所述通用数据对象相关联的数据。

3. 根据权利要求 1 所述的装置，其中，所述数据格式化器按照与所述用户自定义事件相对应的用户自定义数据记录类型来格式化与所述通用数据对象相关联的数据。

4. 根据权利要求 1 所述的装置，其中，所述数据格式化器读取与所述通用数据对象相关联的标签，以确定在格式化与所述通用数据对象相关联的数据时使用的格式。

5. 根据权利要求 1 所述的装置，其中，所述数据格式化器读取与所述通用数据对象相关联的标签，以确定与所述通用数据对象相关联的数据是否为调试数据。

6. 根据权利要求 1 所述的装置，其中，所述数据格式化器读取与所述通用数据对象相关联的标签，以确定与所述通用数据对象相关联的数据是否和与其他通用数据对象相关联的数据相关。

7. 一种对器件测试中生成的自定义事件进行处理的方法，包括：

响应于与对至少一个被测器件的多个测试的执行相对应的事件有序序列，

创建多个数据对象，包括 i) 与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及 ii) 与所述事件中的一个用户自定义事件相对应的通用数据对象；

在分层树结构中使所述数据对象中的一些数据对象与所述数据对象中的其他数据对象相关，同时基于所述用户自定义事件在所述事件有序序列中的位置使所述通用数据对象与所述数据对象中的其他数据对象相关；以及

在所述分层树结构中使与所述事件中的一些事件相对应的数据与所述数据对象中的一些数据对象相关；并且

使多个数据格式化器可以访问与所述分层树结构相关联的所述多个数据对象和数据。

对器件测试中生成的自定义事件进行处理的方法和装置

技术领域

[0001] 本发明涉及用于处理在器件测试期间生成的用户自定义事件的方法和装置。

背景技术

[0002] 在器件特别是电路的测试期间，用于测试该器件的测试程序的执行可能引起多个测试事件以及这些测试事件的属性（如测试结果和其他数据项）的生成。

[0003] 通常，用户必须利用多个预定义的测试事件和属性来创建测试程序。然而有时候，这些预定义的测试事件和属性不允许用户生成其想得到的类型的数据。

发明内容

[0004] 在一个实施例中，一种装置包括数据填充器和数据格式化器。所述数据填充器被提供用来：1）接收与对至少一个被测器件的多个测试的执行相对应的事件有序序列；2）创建多个数据对象，包括与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及与所述事件中的一个用户自定义事件相对应的通用数据对象；3）在分层树结构中使所述数据对象中的一些数据对象与所述数据对象中的其他数据对象相关，基于所述用户自定义事件在所述事件有序序列中的位置使所述通用数据对象与所述数据对象中的其他数据对象相关；以及 4）在分层树结构中使与所述事件中的一些事件相对应的数据与所述的数据对象中的一些数据对象相关。所述数据格式化器被提供用来取回和格式化与所述分层树结构相关联的数据，所述数据格式化器还取回与所述通用数据对象相关联的数据。

[0005] 在另一个实施例中，一种方法包括，响应于与对至少一个被测器件的多个测试的执行相对应的事件有序序列，1）创建多个数据对象，包括 A）与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及 B）与所述事件中的一个用户自定义事件相对应的通用数据对象；2）在分层树结构中使所述数据对象中的一些数据对象与所述数据对象中的其他数据对象相关，基于所述用户自定义事件在所述事件有序序列中的位置使所述通用数据对象与所述数据对象中的其他数据对象相关；以及 3）在所述分层树结构中使与所述事件中的一些事件相对应的数据与所述数据对象中的一些数据对象相关。所述方法还包括使多个数据格式化器可以访问与所述分层树结构相关联的所述多个数据对象和数据。

[0006] 在另一个实施例中，一种装置包括存储在计算机可读介质上的计算机可读代码。所述代码包括代码用来，响应于与对至少一个被测器件的多个测试的执行相对应的事件有序序列，1）创建多个数据对象，包括 A）与所述事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及 B）与所述事件中的一个用户自定义事件相对应的通用数据对象；2）在分层树结构中使所述数据对象中的一些数据对象与所述数据对象中的其他数据对象相关，基于所述用户自定义事件在所述事件有序序列中的位置使所述通用数据对象与所述数据对象中的其他数据对象相关；以及 3）在所述分层树结构中使与所述事件

中的一些事件相对应的数据与所述数据对象中的一些数据对象相关。所述代码还包括代码用来使多个数据格式化器可以访问与所述分层树结构相关联的所述多个数据对象和数据。

[0007] 其他实施例也被公开。

附图说明

[0008] 附图中图解了本发明的说明性实施例，其中；

[0009] 图 1 图示了用于存储和格式化数据的示例性方法；

[0010] 图 2 图示了可作为执行图 1 中示出的方法的结果而被实例化或运行的各种功能单元（或过程）；

[0011] 图 3 图示了图 2 中示出的系统的变体，该系统特别适于在测试环境中使用；

[0012] 图 4 图示了图 3 中示出的系统所使用的 EDL 文件的示例性内容；以及

[0013] 图 5 图示了用于存储图 4 中示出的 EDL 文件的内容的示例性的分层树结构。

具体实施方式

[0014] 作为初步处理方式，注意到在下列描述中，不同附图中出现的相似标号指代相似元件 / 特征。因此将不针对每个附图来对在不同附图中出现的相似元件 / 特征进行描述。

[0015] 为了改善数据被多个数据格式化器格式化的方式，图 1 图示了用于存储将被多个数据格式化器访问的数据的示例性方法 100。方法 100 进行如下。响应于与对至少一个被测器件 (BUT) 的多个测试的执行相对应的事件有序序列，多个数据对象在步骤 102 处被创建。举例来说，在电路测试的领域中，事件可包括下列事件：用信号通知新晶片的装载或卸载；用信号通知对晶片上的特定器件的测试的开始或结束；以及用信号通知不同测试和子测试的开始和结束。

[0016] 方法 100 所创建的数据对象中的至少一些对应于事件中的一些（例如，批次对象、晶片对象、DUT 对象，以及测试和 / 或子测试对象，等等）所稳含的测试结果的逻辑分组。如本描述中所使用的，“隐含”的分组可以是明确提到的那些，或只是推断出的那些。无论如何，逻辑分组最好对应于能被负责格式化数据任务（或负责创建用于格式化数据的数据格式化器的任务）的工程师或用户所理解的真实分组。

[0017] 方法 100 所创建的数据对象中的其他可能是与事件序列中的一个或多个用户自定义事件相对应的通用数据对象。用户自定义事件可表现为各种形式，因此可按照各种方式来配置通用数据对象，如稍后将在本描述中更详细描述的。

[0018] 在创建了数据对象中的一个或多个之后，在步骤 104 处在分层树结构中使数据对象中的一些与数据对象中的其他相关。基于用户自定义事件在事件有序序列中的位置使通用数据对象与数据对象中的其他相关。

[0019] 在步骤 106 中，在分层树结构中使与事件中的一些事件相对应的数据（包括测试结果）与数据对象中的一些相关。

[0020] 虽然分层树结构无需被存储在存储器中，但是这样做将总是有利的，因为这显著加速了数据创建 / 取回过程。

[0021] 在数据对象的创建期间或之后，使若干数据格式化器（即一个或多个数据格式化器）可以访问分层树结构的多个数据对象和数据。见步骤 108。

[0022] 在一个实施例中，数据格式化器中活动的那些中的每一个访问与分层树结构相关联的数据和数据对象，按照数据格式化器所保持的规则来格式化数据，然后将格式化后的数据写入文件。对本说明来说，“活动的”数据格式化器是用户已经选择用来格式化特定数据集的多个“可用的”数据格式化器之一。如果方法 100 被应用于电路测试，则数据格式化器中的一个或多个可将数据写为多条测试纪录。

[0023] 注意到图 1 中示出的方法步骤的次序不是必须的，并且步骤的其他次序，包括步骤的并行处理，是可以的。

[0024] 图 1 中示出的方法 100 可通过存储在计算机可读介质上的计算机可读代码实现。计算机可读介质可包括位于例如单个位置处或分布在网络上的任何数目的固定或可移动介质（如一个或多个硬盘、随机存取存储装置 (RAM)、只读存储装置 (ROM)，或光盘）或其混合。计算机可读代码将通常包括软件，但也可包括固件或程序电路。

[0025] 在一个实施例中，实现了方法 100 的计算机可读代码可使图 2 中示出的功能单元（或过程）被实例化或运行。功能单元包括数据填充器 200 和多个数据格式化器 202、204、206、208，以及其他可选组件。然而，注意到各种功能单元之间的边界是不定的，并且下面所述的某些功能可以通过功能单元中不同的一些来替代执行，或者两个或多个单元的功能可被并入单个功能单元（或过程）。

[0026] 数据填充器 200 用于，1) 接收多个事件 210，2) 在存储器 212 中创建多个数据对象，3) 使与事件中的一些事件相对应的数据与数据对象中的一些相关，以及 4) 将数据存储在存储器 212 中。数据填充器 200 所创建的数据对象中的至少一些数据对象对应于数据填充器 200 所接收到的事件中的一些事件所隐含的数据的逻辑分组。对本公开来说，“隐含”的分组包括被明确“表达”的那些。

[0027] 多个数据格式化器 202、204、206、208 访问数据对象中的一些数据对象，然后将与数据对象有关的数据取回和格式化。

[0028] 在一个实施例中，计算机可读代码还可实例化或运行通知管理器 214。通知管理器 214 可从数据填充器 200 接收事件的指示，并且响应于这些指示，可向数据格式化器 202、204、206、208 中的一个或多个提供事件中的一些事件的通知。数据格式化器 202、204、206、208 可然后被配置为响应于通知而开始它们对数据对象的访问（和数据的取回）。注意到在一些情况下，通知管理器 214 所接收到的事件的指示可对应于比数据填充器 200 所接收的事件更少或不同的事件。例如，在电路测试的情况下，数据填充器 200 可能接收到被认为不够重要故未被抛给通知管理器 214 的指示以及未被用作创建数据对象的基础的测试设置事件的指示。另外，可能有数据填充器 200 从其接收到的事件推断出的事件。例如，基于部件号或其他标记的改变，数据填充器 200 可推断出新的一“批”器件正被测试，然后向通知管理器 214 提供该事件的指示（就是说，即使数据填充器 200 自身可能未接收到新的一“批”事件）。

[0029] 上面所公开的方法 100 和装置可被用于许多应用。在一个具体应用中，方法 100 和装置被用来格式化由安捷伦科技有限公司提供的 93000SOC 系列测试仪生成的测试结果。

[0030] 93000 SOC 系列测试仪（在下文中被称为“93000 测试仪”）是将测试结果和事件记录到通称为 EDL（事件数据记录）文件的二进制数据文件中的 SOC（片上系统）测试仪。该 EDL 文件中的事件对应于对至少一个被测器件 (DUT) 的多个测试的执行，并且被存储在

有序序列中。然而，存储在 EDL 文件中的事件不被“抛”给任何其他过程，并且仅被记录到 EDL 文件。在这一应用中，图 1 中示出的方法 100 还可包括以下步骤 :1) 解析对应于多个事件的数据文件（例如 EDL 文件），以取回所述多个事件，然后 2) 将从所述数据文件取回的事件抛给过程（例如图 2 中示出的数据填充器 200），该过程创建所述多个数据对象并将数据存储在存储器 212 中。

[0031] 在一个实施例中，响应于事件管理器进行的方法调用而解析所述 EDL 文件。如图 3 中示出，事件管理器 300 可对共享库 302 进行方法调用（例如，取事件 ;取事件属性），并且共享库 302 可随后从 EDL 文件 304 取回事件并将它们“抛”给事件管理器 300。事件管理器 300 然后将事件传递给数据填充器 200。

[0032] 共享库 302 可表现为经编译代码的形式，如数据检索库 (DRL)，其在被事件管理器 300 调用时执行一种或多种方法。

[0033] 在存储器 212 中创建的数据对象在分层树结构中被与另一个数据对象相关。作为其他数据对象的子的数据对象可保持指向其父数据对象的指针，而父数据对象无需保持指向所有其子的一列指针。如稍后将在本描述中说明的，这些从子指向其父的指针可以帮助删除不再需要的数据对象的过程。

[0034] 在数据对象是基于多个电路测试的执行的情况下，事件所隐含的数据的逻辑分组可包括诸如批次、晶片和 DUT 的分组这样的一个或多个硬件分组，以及诸如对应于测试和子测试的测试结果的分组这样的一个或多个测试分组。

[0035] 数据可能以各种方式与数据对象相关，包括通过 :1) 直接将数据存储在数据对象中，或者 2) 将数据存储在与数据对象有关（例如，通过指针或其他方式相关）的数据结构中。

[0036] 在 EDL 文件 304 中，数据被存储为事件的属性。因此，如果数据填充器 200 接收到得自 EDL 文件 304 的事件，数据填充器 200 可通过从事件的属性提取数据来提取对应于事件的数据。在电路测试的情况下，被提取的数据可包括测试结果。

[0037] 举例来说，图 4 图示了 EDL 文件 304 的内容的示例性实施例，其中若干数据包括测试结果，并且与记录的事件相关联的若干数据包括用户自定义数据。图 5 图示了数据填充器 200 可从图 4 中示出的 EDL 文件 304 的内容创建的示例性的分层树结构 500。树结构 500 包括两个批次对象 502、504（其中每个可保持指向父“晶片对象”506 的指针）和六个器件对象 508、510、512、514、516、518（其中每个保持指向批次对象 502、504 中的相应一个的指针）。如图所示，可用的测试结果被与器件对象 508、510、512、514、516、518 中的每一个相关联。树结构 500 还包括两个通用对象 520、522，其中的每一个基于 EDL 文件 304（图 4）中的用户自定义事件的位置而与器件对象 508、514 中的特定一个相关。

[0038] 除将各个数据项存储在存储器 212 中之外，数据填充器 200 还可积累诸如测试数据统计量这样的数据统计量，然后使这些数据统计量与数据对象中的一些相关（例如，通过将数据统计量存储在数据对象中，或者通过将数据统计量存储在与数据对象有关的数据结构中）。在一个实施例中，数据统计量可通过数据填充器 200 积累然后在完全编译后与数据对象相关。在一个替代实施例中，不完整的数据统计量可与数据对象相关然后被更新。类似于数据统计量，数据填充器 200 可积累诸如测试数据解释这样的数据解释，并且使它们与数据对象中的一些相关。

[0039] 选择让数据填充器200还是各个数据格式化器202、204、206、208来编译统计量和/或解释数据可以基于不同格式的统计量和解释的有用性。就是说，如果统计量或解释很可能被多个数据格式化器202、204、206、208需要，则常常最好通过数据填充器200一次编译完统计量或解释。另一方面，特定格式的统计量和解释最好利用数据格式化器202、204、206、208中的特定一个来编译。

[0040] 数据格式化器202、204、206、208有许多方式可访问由数据填充器200创建的数据和数据对象。在一个实施例中，数据格式化器202、204、206、208可以仅地监视数据对象。然而，这会需要大量存储装置带宽，并且常常不是很有效。在优选实施例中，数据填充器200生成指向其创建的数据对象的指针，然后将这些指针传递给通知管理器214。通知管理器214随后将这些指针中的一些分发给数据格式化器202、204、206、208中的一些。

[0041] 注意到通知管理器214对于每个数据对象仅需接收指向其的一个指针。数据管理器214可随后为数据格式化器202、204、206、208中的每一个复制该指针，或者向数据格式化器202、204、206、208中的每一个广播该指针。可替代的，通知管理器214可不为数据格式化器202、204、206、208中的每一个复制该指针或向数据格式化器202、204、206、208中的每一个广播该指针，并且可仅为数据格式化器202、204、206、208中的某些复制该指针或向数据格式化器202、204、206、208中的某些广播该指针，所述的某些数据格式化器已经订阅了数据对象所表示的那种类型的数据。

[0042] 通常，数据填充器200的操作将被给予高于数据格式化器202、204、206、208的操作的优先级。为了进一步控制对存储了数据对象和数据的存储器212的访问，数据填充器200（或者创建了所述多个数据对象并将数据存储在存储器212中的其他过程）可被提供对数据对象和数据的直接访问。然而，可经由结构化接口向多个数据格式化器202、204、206、208提供对所述多个数据对象的访问，所述结构化接口协调/仲裁所述数据格式化器对所述数据对象和数据的访问。

[0043] 要减少被保持在存储器212中的数据对象的数目，数据填充器200可监视数据对象的保留，并且可删除不再被任何引用或者访问这些数据对象的对象和过程保留的数据对象。在一个实施例中，当引用数据对象的指针被生成时，认为该数据对象被保留。数据填充器200可在创建数据对象时生成一个这样的指针，然后通知管理器214可生成该指针的额外副本并将其分发给数据格式化器202、204、206、208。当数据对象的子对象被创建时，数据填充器200也可生成指向该对象的指针。当指针被生成时，引用了特定数据对象的指针数目的计数可被保持（可能在数据对象自身中）。数据格式化器202、204、206、208和其他过程可以然后被编程为在其已访问完数据对象时释放它们的指向该数据对象的指针，并且数据填充器200在所有这些指针已被释放之后可将该数据对象删除。由于子对象引用了其父，所以父对象在其所有子对象首先被删除之前无法被删除。

[0044] 图3中示出的数据格式化器可表现为各种形式，包括诸如ASCII（美国信息交换标准代码）格式器202、XML（可扩展标记语言）格式器204、EDL格式器206和/或STDF（标准测试定义格式）格式器208这样的形式。

[0045] 当从存储器212取回通用数据对象时，数据格式化器202、204、206、208中的一个或多个可按照对应于用户自定义事件的用户自定义记录类型来格式化与该通用数据对象相关联的数据。或者，数据格式化器202、204、206、208中的一个或多个可按照通用记录类

型来格式化与通用数据对象相关联的数据。例如，STDF 格式器 208 可按照 STDF 规范所定义的通用数据记录 (GDR) 来格式化与通用数据对象相关联的数据。

[0046] 在一些情况下，通用数据对象可与指定用户自定义对象的类型的标签相关联。该类型可随后被数据格式化器 202、204、206、208 中的一个或多个用来确定在格式化与通用数据对象相关联的数据时使用何种格式。作为示例，标签可规定通用数据对象包括调试数据，或者一种特定的测试数据。

[0047] 在一个实施例中，用户自定义事件和通用数据对象可被用来存储将被连在一起的数据的“片段”。例如，STDF 规范将数据记录限于 64 千字节的数据。在与“事件”相关联的数据可能超过该限制的情况下，用户自定义事件可被定义为将数据集分解为更小的片段。与这些“片段”相关联的标签可随后规定哪部分的数据与特定用户自定义事件相关联，并且通用数据对象可被用来存储数据的所有这些“片段”。一旦数据的片段已经利用例如 STDF GDR 记录来格式化，则下游解析软件或其他过程可以读取这些记录的标签部分，以确定如何将记录的数据连在一起。

[0048] 取决于它们的实现，这里所公开的数据模型（就是说，其中基于数据的逻辑分组创建了数据对象，和 / 或在分层树结构中使数据对象彼此相关联）可以提供许多优势，特别是在电路测试的环境中。例如，通过提供各个过程来将数据组织在存储装置中，然后使多个数据格式化器可以使用该数据，从而从数据格式化器消除了必须读取和组织数据的大部分开销；并且，通过“在存储器中”创建数据对象，它们较之存储在硬盘上的数据可以被更快速地访问。这使数据格式化器的更轻量级，并且提供了可被新数据格式化器有效利用的标准数据模型。这还能够：1) 使得更容易对数据填充器和独立数据格式化器进行修订和修补，2) 使能够对这些功能单元中的每一个进行并行和更快速的编码，以及 3) 减少对每个功能单元进行编码时出错的可能性（例如，因为每个都是轻量级的，所以对每个单元要测试的代码更少）。另外，它使得测试仪能够以在运行时测试环境中更有效率地生成的形式来生成数据，同时使测试数据以更加用户友好和逻辑划分的方式对数据格式化器（和其作者）可用。

[0049] 这里所公开的数据模型，以及将数据填充到数据模型和从数据模型去除数据的方式还有助于平衡：1) 多个数据格式化器可以格式化公共数据集的速度（就是说，它们可以各自并行地格式化数据集的数据）；2) 对存储器、存储装置和 / 或处理资源的实际限制。在电路测试的领域中，特别是当利用 SOC 测试仪来进行参数测试时，如此多的数据被生成，以至于在测试了 10-20 这么少的芯片之后即使良好配置的计算机系统也可达到资源限制。然而，通过利用这里所公开的数据模型以及除去不再需要的数据对象，资源限制通常可被避免。

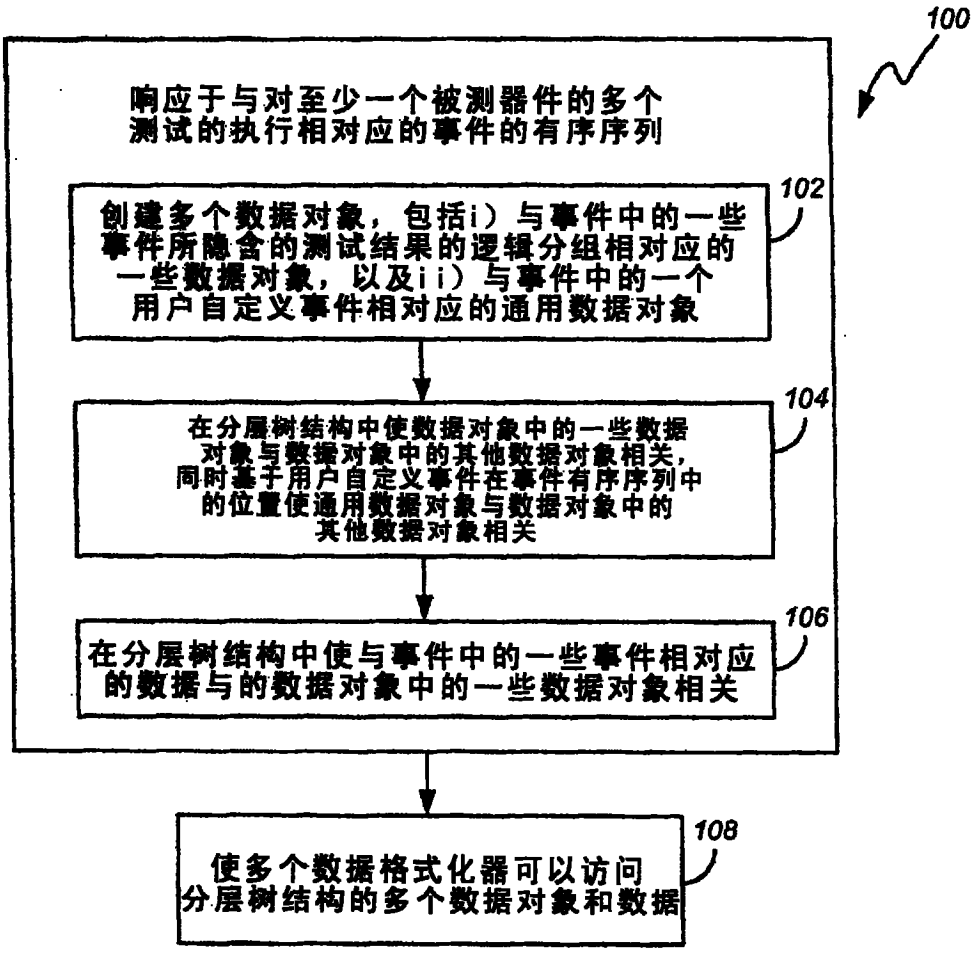

100
响应于与对至少一个被测器件的多个测试的执行相对应的事件的有序序列
102
创建多个数据对象，包括i）与事件中的一些事件所隐含的测试结果的逻辑分组相对应的一些数据对象，以及ii）与事件中的一个用户自定义事件相对应的通用数据对象
104
在分层树结构中使数据对象中的一些数据对象与数据对象中的其他数据对象相关，同时基于用户自定义事件在事件有序序列中的位置使通用数据对象与数据对象中的其他数据对象相关
106
在分层树结构中使与事件中的一些事件相对应的数据与的数据对象中的一些数据对象相关
108
使多个数据格式化器可以访问分层树结构的多个数据对象和数据

图 1

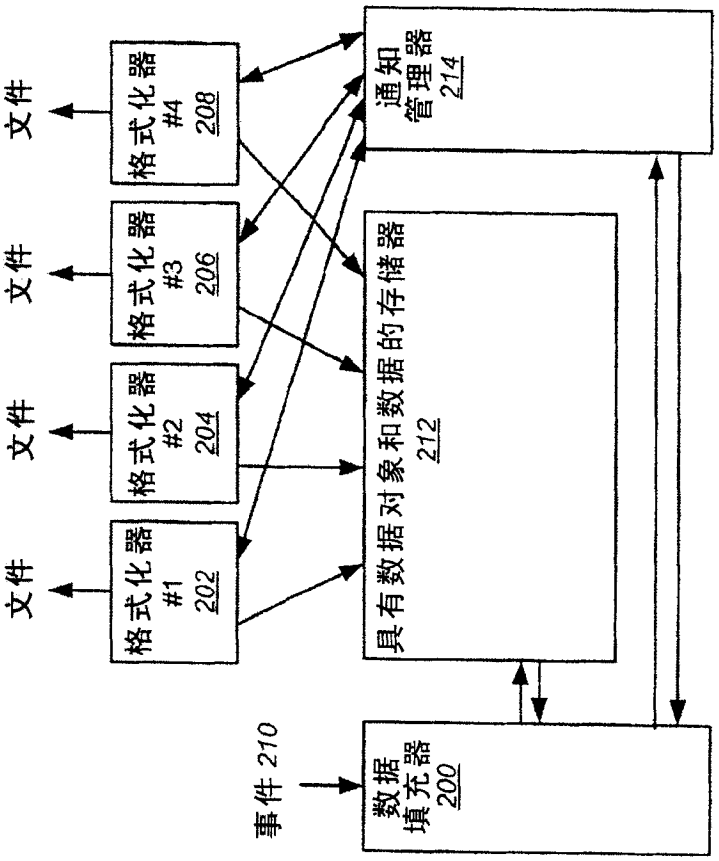

文件
文件
文件
文件
格式化器 #1 202
格式化器 #2 204
格式化器 #3 206
格式化器 #4 208
具有数据对象和数据的存储器 212
通知管理器 214
事件 210
数据填充器 200

图2

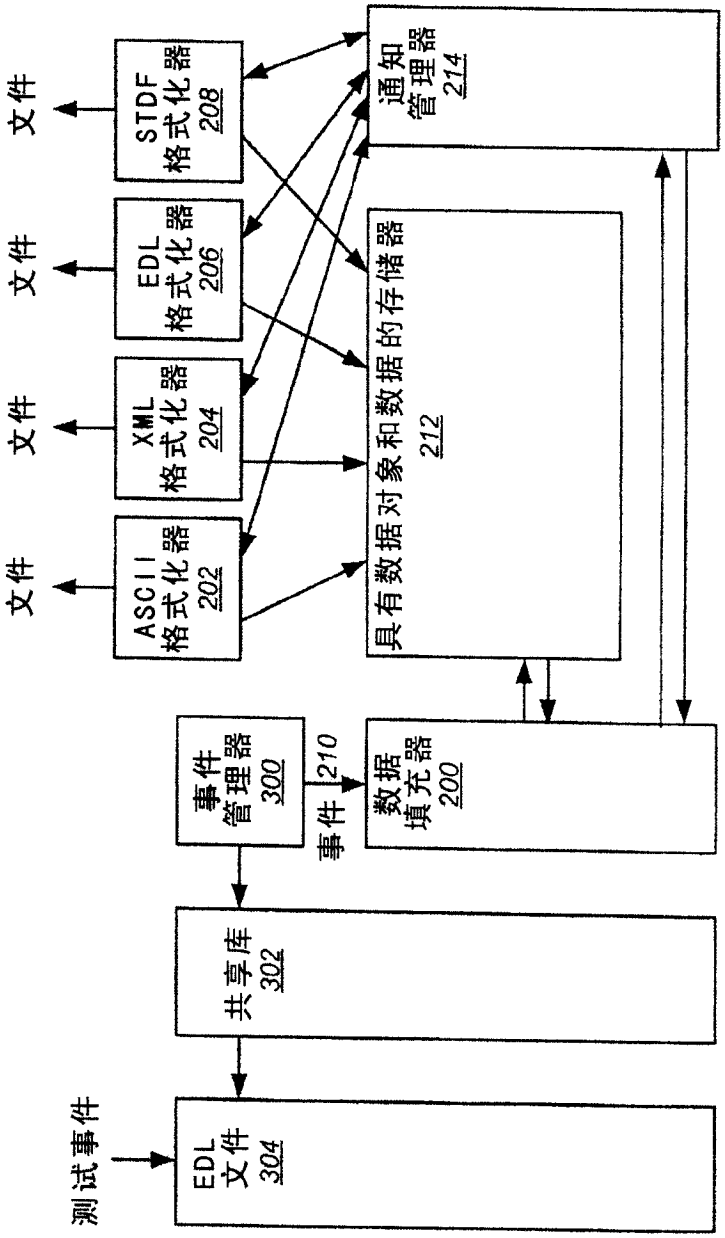

文件
文件
文件
文件
STDF 格式化器 208
EDL 格式化器 206
XML 格式化器 204
ASCII 格式化器 202
通知管理器 214
具有数据对象和数据的存储器 212
事件管理器 300
210
事件
数据填充器 200
共享库 302
测试事件
EDL 文件 304

图3

行	记录的事件	数据
1	Start Lot Level	
2	Variable Assignment	Lot_id=1
3	Start Device Level	
4	Variable Assignment	Device_id=a
5	Device tested	<test results for device "a">
6	Start User-defined Level	
7	User-defined event	<data>
8	End User-defined Level	
9	Variable Assignment	Device_id=b
10	Device tested	<test results for device "b">
11	Variable Assignment	Device_id=c
12	Device tested	<test results for device "c">
13	End Device Level	
14	End Lot Level	
1	Start Lot Level	
2	Variable Assignment	Lot_id=2
3	Start Device Level	
4	Variable Assignment	Device_id=d
5	Device tested	<test results for device "d">
6	Start User-defined Level	
7	User-defined event	<data>
8	End User-defined Level	
9	Variable Assignment	Device_id=e
10	Device tested	<test results for device "e">
11	Variable Assignment	Device_id=f
12	Device tested	<test results for device "f">
13	End Device Level	
14	End Lot Level	

图 4

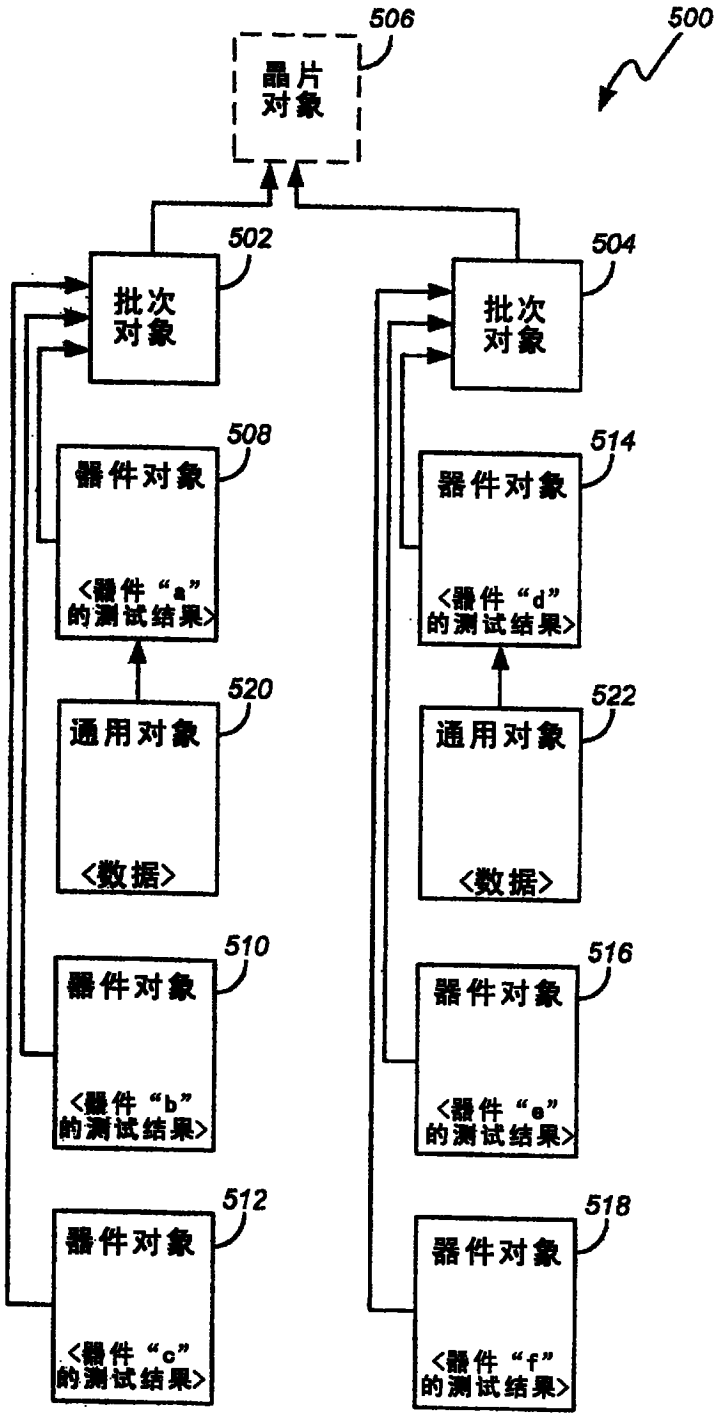

506
500
晶片对象
502
批次对象
504
批次对象
508
器件对象
<器件“a”的测试结果>
514
器件对象
<器件“d”的测试结果>
520
通用对象
<数据>
522
通用对象
<数据>
510
器件对象
<器件“b”的测试结果>
516
器件对象
<器件“e”的测试结果>
512
器件对象
<器件“c”的测试结果>
518
器件对象
<器件“f”的测试结果>

图 5