

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680012050.5

[43] 公开日 2008 年 4 月 9 日

[11] 公开号 CN 101160583A

[22] 申请日 2006.3.17

[21] 申请号 200680012050.5

[30] 优先权

[32] 2005.4.14 [33] US [31] 11/106,412

[86] 国际申请 PCT/EP2006/060823 2006.3.17

[87] 国际公布 WO2006/108753 英 2006.10.19

[85] 进入国家阶段日期 2007.10.12

[71] 申请人 国际商业机器公司

地址 美国纽约

[72] 发明人 R·A·霍特 J·W·米勒

J·拉米雷斯

[74] 专利代理机构 北京市中咨律师事务所

代理人 于 静 李 峥

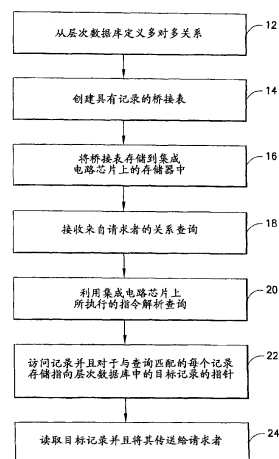
权利要求书 2 页 说明书 6 页 附图 4 页

[54] 发明名称

层次数据库的关系查询

[57] 摘要

快速处理对具有数据表的层次数据库的关系查询。创建桥接表以便将多对多关系转换成多个一对多关系。所述桥接表存储在定制半导体芯片上，其解析所述查询并且通过使用所述桥接表，利用存储在所述半导体芯片上的定制数据算法来确定所述数据表中的哪些记录与所述查询匹配。对于每个匹配，存储指向所述层次数据库中的目标记录的指针。在定制芯片上执行的指令读取所述指针或目标记录，并且将其传送给查询请求者。



1. 一种查询层次数据库的方法，其包括以下步骤：

为所述层次数据库定义多个多对多关系；

创建具有记录的桥接表，以便将第一与第二实体之间的所述多对多关系转换成所述第一实体与所述桥接表之间的一对多关系，以及所述桥接表与所述第二实体之间的一对多关系；

将所述桥接表存储在集成电路芯片中的存储器中；

接收来自请求者的关系查询；

通过所述集成电路芯片上所执行的指令来解析所述关系查询；

访问所述桥接表中所述记录中的每一个，并且如果所述每个记录均符合所述查询，则存储指向所述层次数据库中的目标记录的指针；以及

读取所有的所述目标记录或所述指针，并且将所读取的目标记录或指针传送给所述请求者。

2. 根据权利要求1的方法，其中所述多对多关系在实体之间，所述实体包括所述实体中特定的一个与所述实体中另一个之间的互连。

3. 根据权利要求2的方法，其进一步包括步骤：使用所述桥接表来创建连接表，所述连接表描述了所述实体中所述特定的一个与所述实体中所述另一个之间的所述互连。

4. 根据权利要求1的方法，其中所述指令存储在所述互连芯片上。

5. 根据权利要求1的方法，其中所述关系查询是SQL查询。

6. 根据权利要求1的方法，其中使用定制寻址算法来读取所述目标记录。

7. 一种用于查询层次数据库的系统，其包括：

这样的装置，其用于为所述层次数据库定义多个多对多关系；

集成电路芯片，其具有存储器和指令处理器；

这样的装置，其用于创建存储在所述集成电路芯片上的桥接表，所述桥接表具有记录，以便将第一与第二关系之间的所述多对多关系转换成所

述第一关系与所述桥接表之间的一对多关系，以及所述桥接表与所述第二关系之间的一对多关系；

这样的装置，其用于接收来自请求者的关系查询；

这样的装置，其用于通过所述集成电路芯片上的所述指令处理器所执行的指令来解析所述关系查询；

这样的装置，其用于访问所述桥接表中所述记录中的每一个并且如果所述每个记录均符合所述查询，则存储指向所述层次数据库中的目标记录的指针；以及

这样的装置，其用于读取所有的所述目标记录或所述指针，并且将所读取的目标记录或所述指针传送给所述请求者。

8. 根据权利要求7的系统，其中所述多对多关系在实体之间，所述实体包括所述实体中特定的一个与所述实体中另一个之间的互连。

9. 根据权利要求8的系统，其进一步包括用于使用所述桥接表来创建连接表的装置，所述连接表描述了所述实体中所述特定的一个与所述实体中所述另一个之间的所述互连。

10. 根据权利要求7的系统，其中所述指令存储在所述互连芯片上。

11. 根据权利要求7的系统，其中所述关系查询是SQL查询。

12. 根据权利要求7的系统，其中使用定制寻址算法来读取所述目标记录。

13. 一种可载入数字计算机的内部存储器的计算机程序产品，其包括：当在计算机上运行所述产品时用于执行以实现根据权利要求1至6所述的本发明的软件代码部分。

层次数据库的关系查询

技术领域

本发明涉及用于访问层次数据库（**hierarchical database**）的方法和系统，并且特别涉及使用关系查询来访问这些数据库。甚至更特别地，本发明涉及用于在接收到来自请求者的关系查询之后快速访问层次数据库并且将该关系查询所标识的那些目标记录快速返回给请求者的机制。

背景技术

很多公司和政府使用层次数据库来获取和检索与交易（特别是由公司或政府主体所进行的商业交易）关联的数据。层次数据库使用层次模式（**hierarchical schema**）来存储被称为父/子模型的信息。层次模式可以表示为树状结构，其中每个父节点可以具有多个子节点，而每个子节点只可以具有一个父节点。

另一常用的数据库是关系数据库，其是具有被这样定义的数据的表格数据库，即可以以多种不同的方式重新组织和访问该数据。在关系数据库中，数据记录被维护于数据表中或者全部具有相同的列的行的集合中。每行均是数据记录，并且每列保留该数据记录的特定类型的数据的信息。可以使用将不同表格中不同的数据记录连接到一起的唯一的索引或关键字来索引数据记录。

关系数据库特别有用，因为可以使用关系查询语言来访问存储在其中的信息。一种这样的查询语言，SQL（结构化查询语言）被关系数据库用户广泛使用和理解，SQL是国际商业机器公司的商标。

不幸的是，在不提供除了正常可用的之外的附加能力的情况下，询问层次数据库的基本关系问题是不可能的。例如，很多公司和其它组织通过维护两个数据库（层次数据库和关系数据库）以及相关的支持人员来支持

其操作。维护该方法成本高昂并且麻烦。

Hoth 等人在美国专利申请 2004/0030716 A1 中描述了一种用于在层次数据库中提供关系模式的方法。创建桥接表 (bridging table) 以描述和证明层次数据库中的实体之间的互连。在此通过引用的方式将上述 Hoth 的专利申请完全并入本说明书。

虽然 Hoth 所描述的方法提供了所期望的能力,但是由于在形成桥接表以及在将查询响应传递给客户时需要开销,因此该方法在响应时间方面常常是缓慢的。

需要改进查询响应时间以满足顾客对利用其层次数据库的查询能力的需求。

发明内容

因此,本发明的首要目的是提供一种从层次数据库快速提供对客户查询的响应数据的方法,其以关系查询语言呈现。

另一目的是提供一种具有这样的快速响应查询能力的系统。

依照本发明的一个实施例达到了这些和其它的目的,在该实施例中提供了一种查询层次数据库的方法,其包括以下步骤:为所述层次数据库定义多个多对多关系;创建具有记录的桥接表,以便将第一与第二实体之间的所述多对多关系转换成所述第一实体与所述桥接表之间的一对多关系,以及所述桥接表与所述第二实体之间的一对多关系;将所述桥接表存储在集成电路芯片中的存储器中;接收来自请求者的关系查询;通过所述集成电路芯片上所执行的指令来解析所述关系查询;访问所述桥接表中的每个记录,并且如果所述每个记录均符合所述查询,则存储指向所述层次数据库中的目标记录的指针;以及读取所有的所述目标记录或所述指针,并且将所读取的目标记录或指针传送给所述请求者。

依照本发明的另一实施例,提供了一种用于查询层次数据库的系统,其包括:用于为所述层次数据库定义多个多对多关系的装置;具有存储器和指令处理器的集成电路芯片;用于创建存储在所述集成电路芯片上的桥

接表的装置，所述桥接表具有记录以便将第一与第二关系之间的所述多对多关系转换成所述第一关系与所述桥接表之间的一对多关系，以及所述桥接表与所述第二关系之间的一对多关系；用于接收来自请求者的关系查询的装置；用于通过所述集成电路芯片上的指令处理器所执行的指令来解析所述关系查询的装置；用于访问所述桥接表中的每个记录并且如果所述每个记录均符合所述查询，则存储指向所述层次数据库中的目标记录的指针的装置；以及用于读取所有的所述目标记录或所述指针并且将所读取的目标记录或指针传送给所述请求者的装置。

附图说明

图 1 是说明了本发明的步骤的流程图；

图 2 是描绘了用于提供桥接表的方法的流程图；

图 3 是适于本发明的半导体芯片的功能框图；以及

图 4 是说明了对目标记录的读取的流程图。

具体实施方式

在图 1 中，依照本发明示出了描绘实现查询层次数据库的改进方法所需要的步骤的流程图。在步骤 12 中，为层次数据库定义多个多对多关系。如以上所指出的，层次数据库支持一对多关系。可以使用任何类型的层次数据库，例如在 LOTUS NOTES 软件产品中所使用的数据库（LOTUS NOTES 是马萨诸塞州 Lotus Development Corporation of Cambridge 的注册商标）。

因此，为了与层次数据库结构兼容，必须将来自步骤 12 的多对多关系转换成一对多关系。转换方法之一是通过使用在步骤 14 中所创建的桥接表。桥接表是这样构成的，即替换多对多关系，例如，利用第一实体与桥接表之间的多个一对多关系，以及桥接表与第二实体之间的第二种多个一对多关系。

图 2 是描绘了这样的转换过程的流程图。在步骤 32 中，标识关系中的

主要实体。在步骤 34 中，创建概念模式来表示不同的标识实体之间如何彼此相关。因此，该概念模式表示实体关系图。在步骤 36 中，从概念模式创建内部模型。该内部模型标识具有多对多关系的实体，该关系必须被转换以便在层次数据库中使用。

在步骤 38 中，将内部模型变换成适于基本数据库的要求的物理模型。例如，如果该基本数据库是 LOTUS NOTES 的一部分，则可以创建表单和视图。然后，可以将数据输入到表单以填充相应的表格来创建基本数据结构。数据还可以存储在基本数据表中。

在步骤 40 中，使用连接表（joining table）登记互连实体之间的互连。连接表可以包括在步骤 32 中所标识的实体之间的路径，包括使用桥接表互连的实体之间的那些路径。连接表中的每个条目均是从步骤 36 中所创建的内部模型中导出的。连接表中的每个条目均定义了可如何从另一实体出发检索与特定实体关联的数据。

在步骤 42 中，元数据是为每个实体创建的，其定义了可以从相应实体提取的数据的类型。可以检索数据用于显示或呈现给用户。生成元数据包括生成证明实体的表格、实体之间的互连，或者实体之间的数据流。

Hoth 等人在美国专利申请 U.S. 2000/0030716 A1 中进一步详细描述了以上对图 2 所列出的所有步骤。在步骤 44 中，创建用于从层次数据库中搜寻和检索数据的组件。下面结合图 1、3 和 4 来描述该步骤。

在图 3 中，示出了定制半导体芯片 52，其具有处理器核 54、高速缓存 56、闪存 58 和服务协议处理器 60。存储区域网形式的外部存储器 62 附于半导体芯片 52，并且可从半导体芯片 52 对其进行访问。在不背离本发明的情况下，出于其它目的可以在半导体芯片 52 上提供其它的元件。

处理器核 54 包括用于执行程序指令的指令处理器。例如，处理器核可以执行用于解析关系查询的指令，或者用于读取目标记录的指令。可以将指令存储在半导体芯片 52 上。例如，可以将指令存储在高速缓存 56 或闪存 58 中，或者处理器核 54 自身内。此外，还可以将指令存储在存储器 62 上并且根据需要检索该指令来实施本发明。将经常执行的指令存储在高速

缓存 56 中或处理器核 54 自身内。可以将不经常执行的指令存储在闪存 58 或存储器 62 中。半导体设计领域的普通技术人员可以认识到，在不背离本发明的精神的情况下可以在数据存储方面进行这样的折衷和优化。

可以单个或结合其它芯片将半导体芯片 52 安装到常规或专门的单个或多个芯片、芯片载体上。在优选的实施例中，将芯片载体安装到主机箱中用于定位的插件卡上。插件卡优选地适于通过带状电缆或其它装置提供到硬驱动器阵列的连接，并且包括到主机箱内 I/O 总线的连接。

回到图 1，在步骤 16 中，将桥接表存储在图 3 的定制集成电路芯片上的存储器中。在优选的实施例中，将桥接表存储在高速缓存 56 或闪存 58 中。还可以将数据表地址、元数据和表互连存储在高速缓存 56 或闪存 58 中。

在步骤 18 中，从请求者接收关系查询。请求者可以是使用 SQL 查询语言来阐明其关系查询的用户。用户通常希望能够使用对数据库的关系查询来询问商业智能问题。可以在工作站处输入查询，在该工作站上，层次数据库、定制半导体芯片以及本发明的所有其它的软件和硬件元件都是自主式的。更典型地，虽然数据库位于服务器计算机上，但是，用户却在远程连接的工作站、终端设备、膝上型计算机、掌上设备、蜂窝电话或其它便携式设备处输入其查询。

关系查询经互连到达定制半导体芯片 52，在此，通过处理器核 54 中所执行的指令在步骤 20 中解析该关系查询。在步骤 22 中，解析允许该查询与所存储的桥接表或数据表中的记录相匹配。对于符合经解析的查询的每个记录，均存储指针指向层次数据库中的目标记录。可以将指针存储在半导体芯片 52 上的任何地方。例如，可以将指针存储在半导体芯片 52 的高速缓存 56 或闪存 58 中的堆栈中。还可以将指针存储到芯片 52 的外部，例如，存储区域网 (SAN) 62 中，或者任何的存储介质位置中。可以在处理器核 54 中执行用于访问桥接和数据表记录并且将每个记录与经解析的查询进行比较的指令。

在步骤 24 中，可以将堆栈的内容（例如，来自步骤 22 的指针）返回

给请求者。在步骤 24 中，处理器核 54 还可以从层次数据库读取目标记录，并且将这些记录传送给请求者。层次数据库可以位于硬驱动器上或 SAN 62 上。当请求者位于远端时，通常会将目标记录发送回请求的设备，然而，对此并不作要求。本领域中的技术人员可以认识到，一旦检索到目标记录，就可以将该目标记录进一步处理成报告，并且可以将报告或目标记录本身传送给在任何所期望的位置的请求者。

从层次数据库读取大量的目标记录可能是耗时、性能受限的操作。因此，在步骤 22 和 24 中可以使用如图 4 所示的定制寻址算法。在步骤 72 中，通过提供数据库描述符，该算法可以用于任何的层次结构。待读取的记录是基于目标记录本身以及在步骤 72 中所提供的该目标记录的路径。获取并解释数据库描述符以提供到目标段的路径。

在步骤 74 中，建立路径以链接至数据库所在的硬件。例如，如果数据库位于硬驱动器上，那么在步骤 74 中初始化到硬驱动器控制单元的通信链路。

在步骤 76 中，使用以上所描述的数据表地址、桥接表、元数据和表互连来完成一路上（all the way）到目标段的层次读取。

在步骤 78 中，将目标段传送给请求者。如对步骤 24 所指出的，可选地，可以将指针返回给请求者。在步骤 80 中，寻址算法终止。

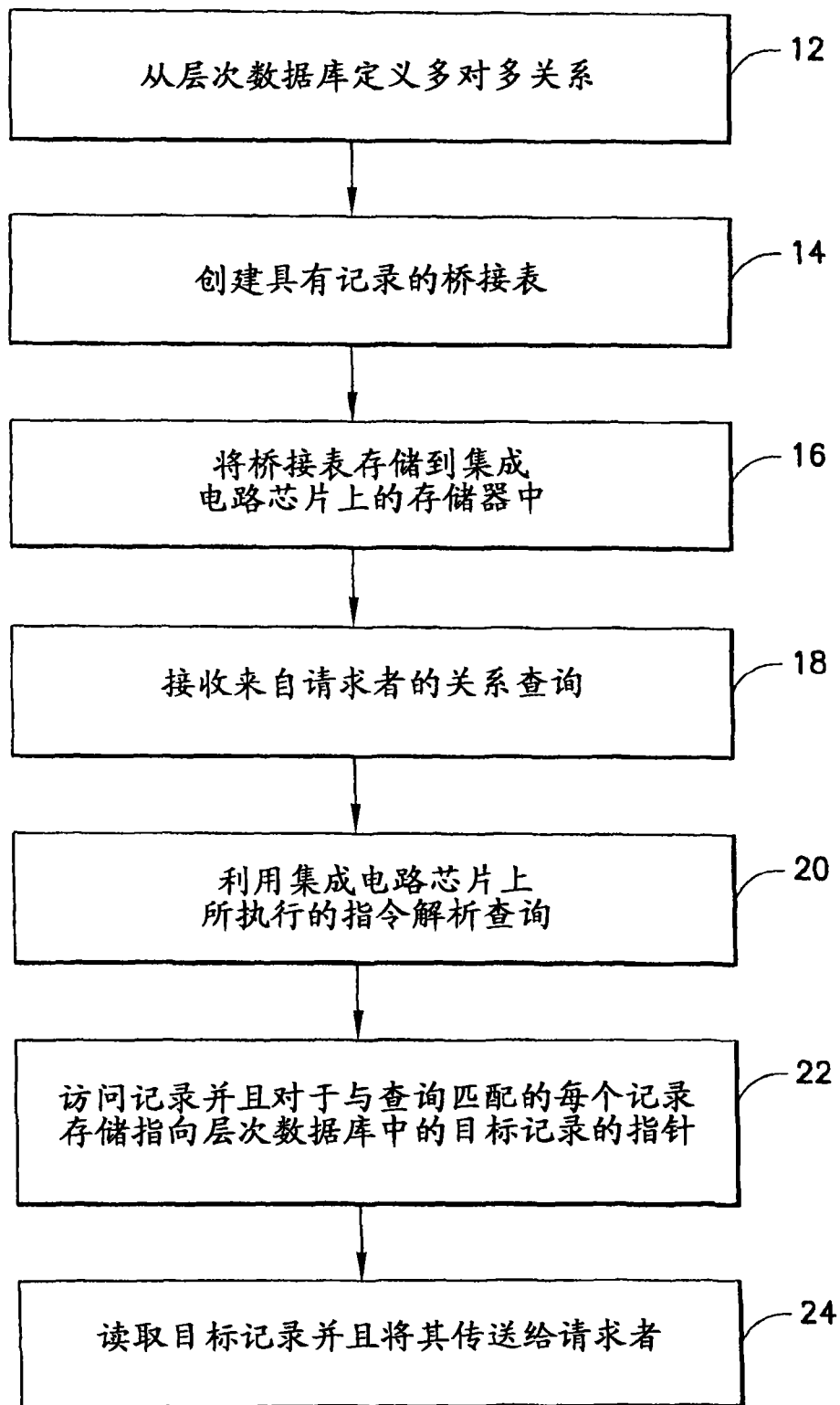


图 1

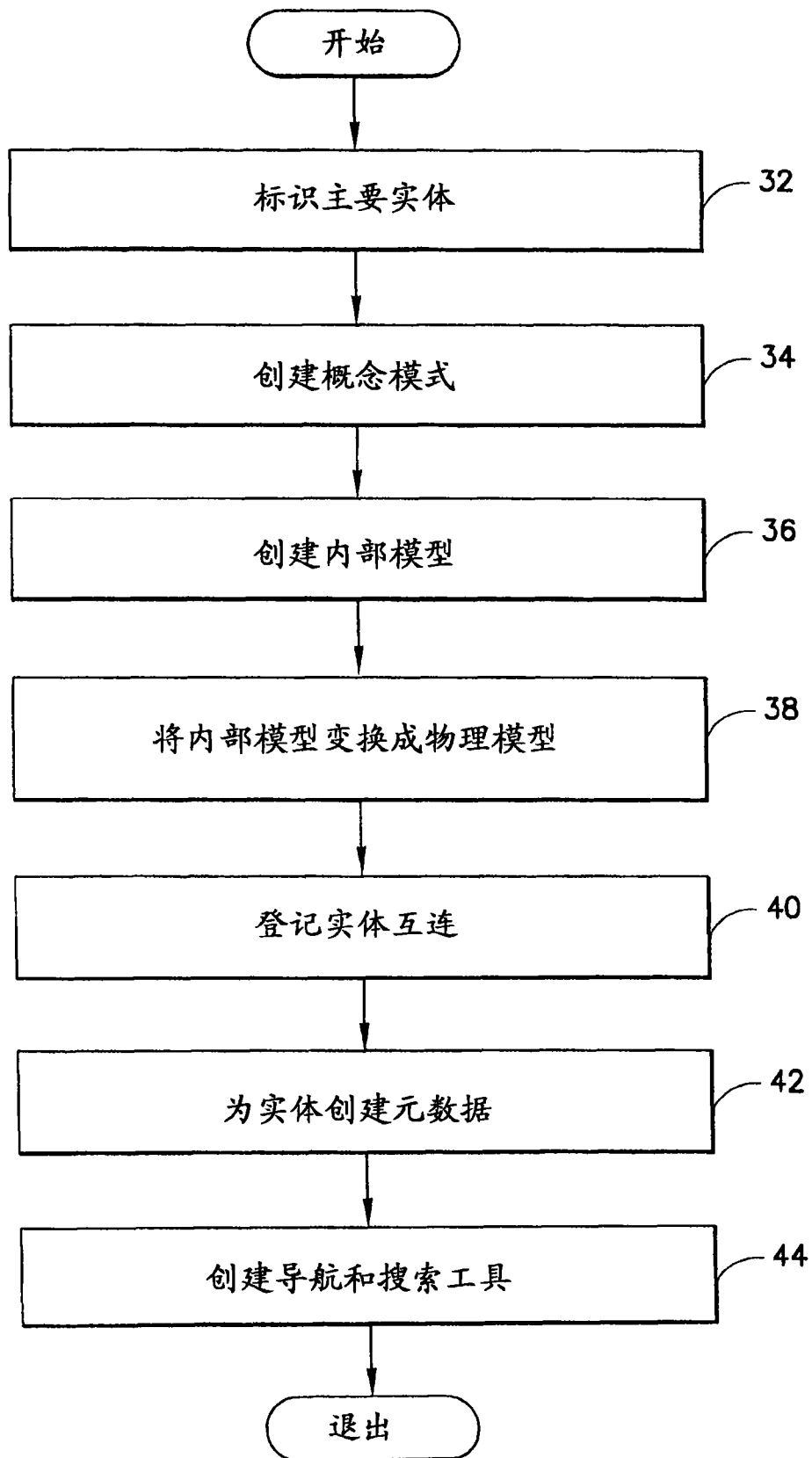


图 2

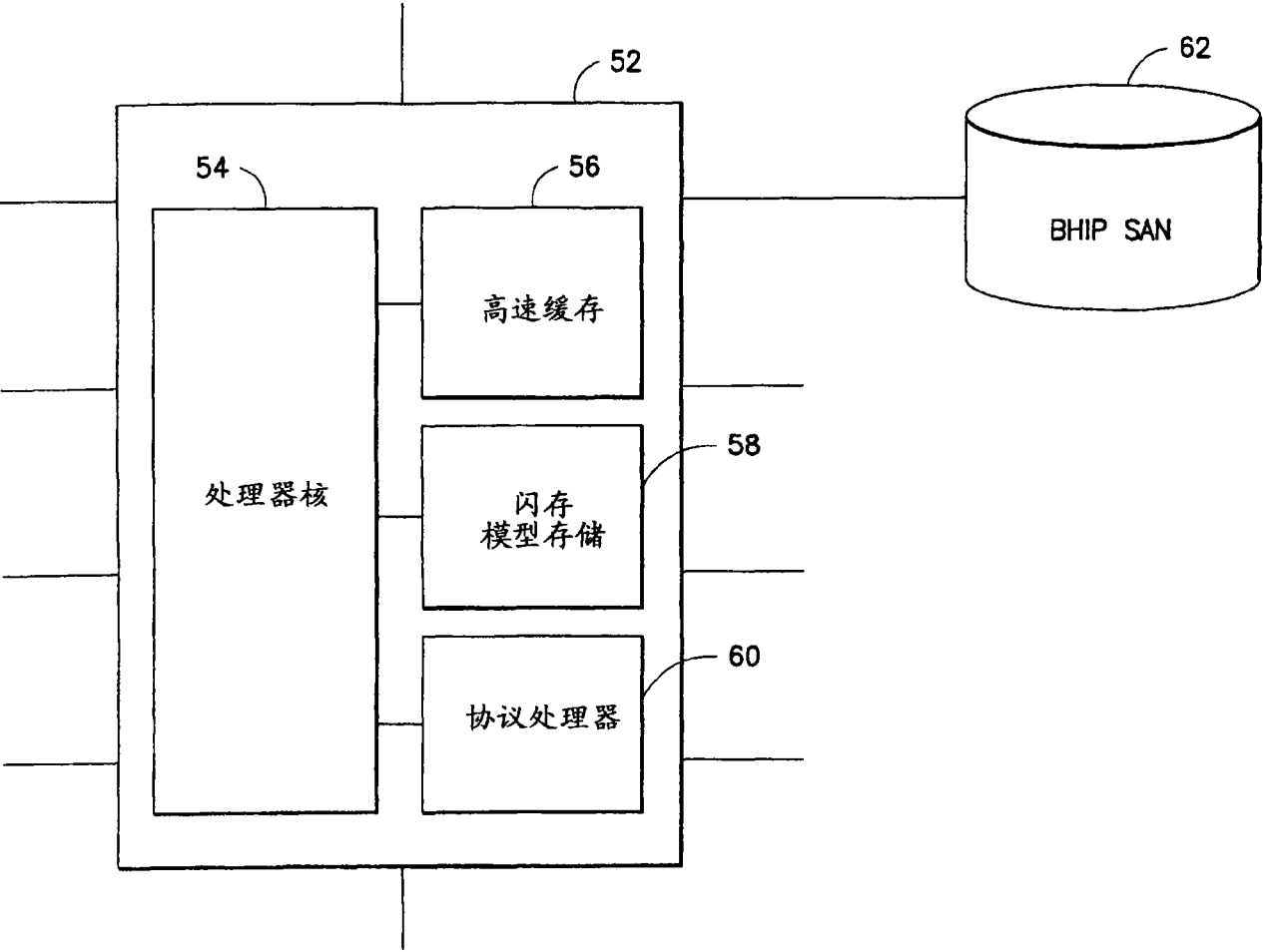


图 3

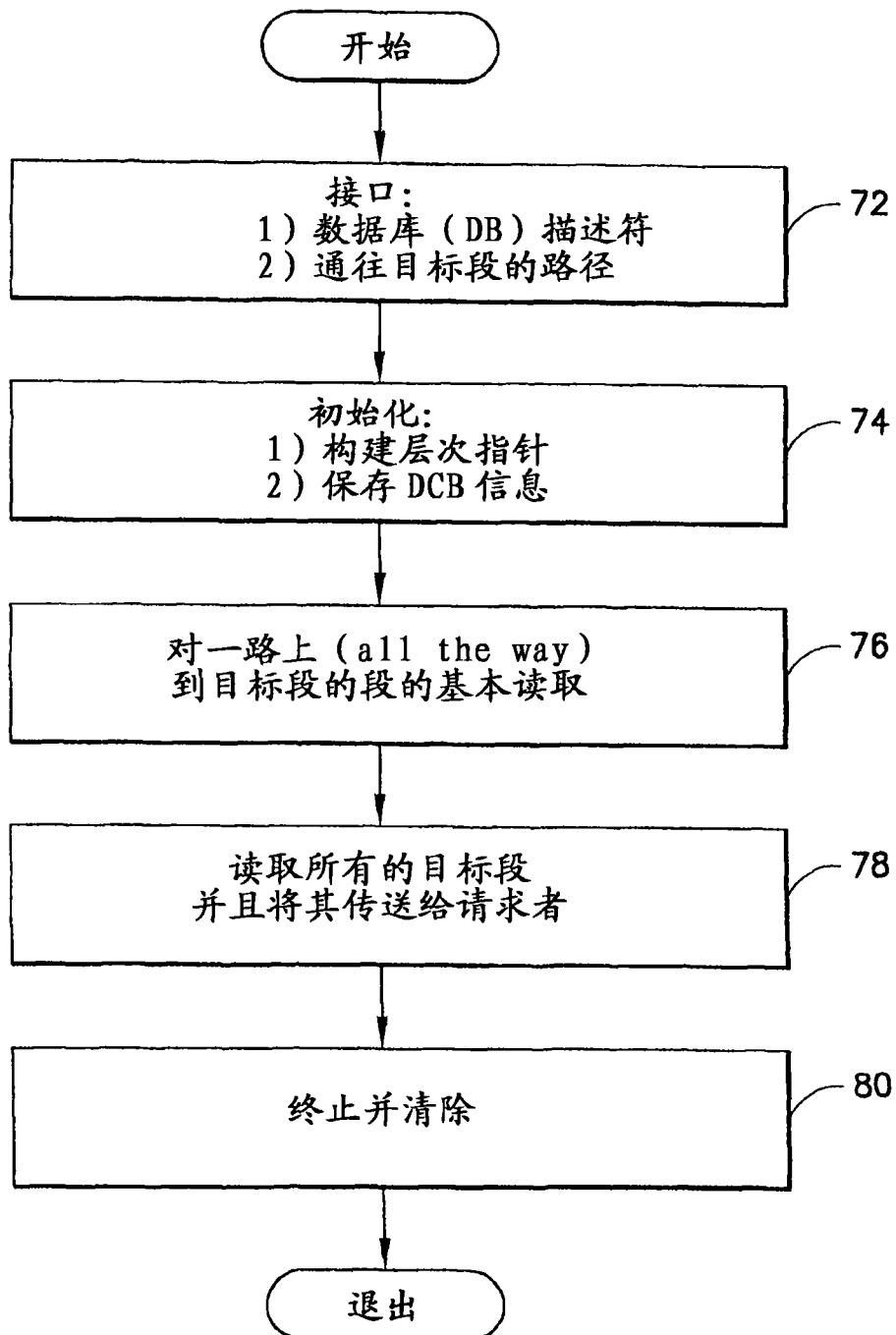


图 4