



(12) 发明专利

(10) 授权公告号 CN 108595295 B

(45) 授权公告日 2023. 09. 12

(21) 申请号 201810380360.7

(22) 申请日 2018.04.25

(65) 同一申请的已公布的文献号
申请公布号 CN 108595295 A

(43) 申请公布日 2018.09.28

(66) 本国优先权数据
201711447642.6 2017.12.27 CN

(73) 专利权人 贵阳忆芯科技有限公司
地址 550081 贵州省贵阳市观山湖区林城
西路摩根中心A座第210层1-20号

(72) 发明人 胡旭 谈笑 孙唐 林岗 郑先翔

(74) 专利代理机构 北京卓特专利代理事务所
(普通合伙) 11572
专利代理师 段宇

(51) Int.Cl.

G06F 11/22 (2006.01)

(56) 对比文件

CN 102902906 A, 2013.01.30

CN 1979686 A, 2007.06.13

US 2003037295 A1, 2003.02.20

US 2009178021 A1, 2009.07.09

US 2009271139 A1, 2009.10.29

US 2014282407 A1, 2014.09.18

US 2011078521 A1, 2011.03.31

郝焱, 汪东升. 嵌入式微处理器的软硬件协同验证. 计算机工程与设计. 2004, (第07期), 1053-1058.

审查员 易建琼

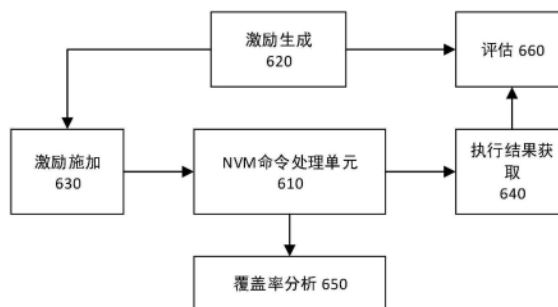
权利要求书2页 说明书12页 附图9页

(54) 发明名称

微指令序列的测试方法与系统

(57) 摘要

提供了微指令序列的测试方法与系统。所提供的测试方法,包括:生成访问NVM芯片的一个或多个命令作为测试微指令序列的激励;获取微指令序列的执行轨迹以及同执行轨迹对应的逻辑单元;根据执行轨迹对应的逻辑单元将执行轨迹分组;从各组执行轨迹中提取同所述微指令序列可能的执行路径匹配的作为被覆盖的执行路径;累计从每组执行轨迹中提取的多个被覆盖的执行路径,得到所有被覆盖的执行路径,通过所有被覆盖的执行路径与所有可能的执行路径之比,得到执行路径覆盖率。



1. 微指令序列的测试方法,包括:

生成测试微指令序列的激励;

获取微指令序列的执行轨迹;

从执行轨迹中提取同所述微指令序列可能的执行路径匹配的作为被覆盖的执行路径;

累计从执行轨迹中提取的多个被覆盖的执行路径,得到执行路径覆盖率;

其中,获取的微指令序列的执行轨迹包括获取按执行顺序的微指令地址序列;

根据微指令序列生成有向图,其中所述有向图包括节点和节点之间的边,节点指示微指令或者微指令所对应的存储器地址,节点之间的边指示了节点所对应的微指令的执行顺序;遍历有向图,获取从有向图的根节点到叶节点的所有可能的执行路径,作为微指令序列所有可能的执行路径;

从执行轨迹中获取同有向图的根节点对应的地址相同的第一存储器地址,从执行轨迹中获取第一存储器地址后续的第二存储器地址,从所有可能的执行路径中搜索同有向图的根节点对应的下一节点对应的地址同第二存储器地址相同的一个或多个执行路径,作为候选执行路径,直到从执行轨迹获取的后续存储器地址,同候选执行路径之一的对应于叶节点的地址相同,以得到匹配的执行路径。

2. 一种存储控制器执行的微指令序列的测试方法,包括:

生成访问NVM芯片的一个或多个命令作为测试微指令序列的激励;

获取微指令序列的执行轨迹以及同执行轨迹对应的逻辑单元;

根据执行轨迹对应的逻辑单元将执行轨迹分组;

从各组执行轨迹中提取同所述微指令序列可能的执行路径匹配的作为被覆盖的执行路径;

累计从每组执行轨迹中提取的多个被覆盖的执行路径,得到所有被覆盖的执行路径,通过所有被覆盖的执行路径与所有可能的执行路径之比,得到执行路径覆盖率;

其中,获取微指令序列的执行轨迹包括获取按执行顺序的微指令地址序列,以及同每个微指令地址相关联的执行该微指令所访问的逻辑单元标识;

根据微指令序列生成有向图,其中所述有向图包括节点和节点之间的边,节点指示微指令或者微指令所对应的存储器地址,节点之间的边指示了节点所对应的微指令的执行顺序;遍历有向图,获取从有向图的根节点到叶节点的所有可能的执行路径,作为微指令序列所有可能的执行路径;

从执行轨迹中获取同有向图的根节点对应的地址相同的第一存储器地址,从执行轨迹中获取第一存储器地址后续的第二存储器地址,从所有可能的执行路径中搜索同有向图的根节点对应的下一节点对应的地址同第二存储器地址相同的一个或多个执行路径,作为候选执行路径,直到从执行轨迹获取的后续存储器地址,同候选执行路径之一的对应于叶节点的地址相同,以得到匹配的执行路径。

3. 根据权利要求2所述的方法,还包括:

若执行路径覆盖率未达到指定阈值,根据所有可能的执行路径中尚未被覆盖的执行路径生成第二激励,使得执行被测试的微指令序列以处理同生成的第二激励对应的命令时,执行被测试的微指令序列的所述尚未被覆盖的执行路径。

4. 根据权利要求2或3所述的方法,其中

访问不同逻辑单元的作为激励的多个命令,被并发地执行。

5. 根据权利要求2或3之一所述的方法,其中

随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

6. 根据权利要求5所述的方法,其中

通过指定生成的多种类型的命令的每种类型的频度与顺序,和/或应用于所生成的命令的一个或多个约束条件而随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

7. 一种存储控制器执行的微指令序列的测试系统,包括:

激励生成模块,用于生成访问NVM芯片的一个或多个命令作为测试微指令序列的激励;

激励施加模块,获取激励生成模块产生的激励,并施加给被测试的存储控制器;

覆盖率分析模块,从所述存储控制器获取所述微指令序列的执行轨迹,从执行轨迹中提取同所述微指令序列可能的执行路径匹配的作为被覆盖的执行路径,并累计从执行轨迹中提取的多个被覆盖的执行路径,得到执行路径覆盖率;

其中,获取的微指令序列的执行轨迹包括获取按执行顺序的微指令地址序列;

根据微指令序列生成有向图,其中所述有向图包括节点和节点之间的边,节点指示微指令或者微指令所对应的存储器地址,节点之间的边指示了节点所对应的微指令的执行顺序;遍历有向图,获取从有向图的根节点到叶节点的所有可能的执行路径,作为微指令序列所有可能的执行路径;

从执行轨迹中获取同有向图的根节点对应的地址相同的第一存储器地址,从执行轨迹中获取第一存储器地址后续的第二存储器地址,从所有可能的执行路径中搜索同有向图的根节点对应的下一节点对应的地址同第二存储器地址相同的一个或多个执行路径,作为候选执行路径,直到从执行轨迹获取的后续存储器地址,同候选执行路径之一的对应于叶节点的地址相同,以得到匹配的执行路径。

8. 根据权利要求7所述的测试系统,还包括:

评估单元,从所述存储控制器获取激励的执行结果,从所述激励生成模块获取激励,并比较执行结果与激励的预期结果。

9. 根据权利要求7或8所述的测试系统,其中

所述覆盖率分析模块,还获取同执行轨迹对应的逻辑单元,根据执行轨迹对应的逻辑单元将执行轨迹分组;从各组执行轨迹中提取同所述微指令序列可能的执行路径匹配的作为被覆盖的执行路径。

10. 根据权利要求7或8之一所述的测试系统,其中

所述激励生成模块通过指定生成的多种类型的命令的每种类型的频度与顺序,和/或应用于所生成的命令的一个或多个约束条件而随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

微指令序列的测试方法与系统

技术领域

[0001] 本申请涉及存储技术,尤其涉及对存储控制芯片的介质接口控制器进行测试的方法与装置,以加快以覆盖率为目标的测试过程。

背景技术

[0002] 参看图1,展示了存储设备的框图。存储设备102同主机相耦合,用于为主机提供存储能力。主机同存储设备102之间可通过多种方式相耦合,耦合方式包括但不限于通过例如SATA、IDE、USB、PCIE、NVMe (NVM Express)、SAS、以太网、光纤通道、无线通信网络等连接主机与存储设备102。主机可以是能够通过上述方式同存储设备相通信的信息处理设备,例如,个人计算机、平板电脑、服务器、便携式计算机、网络交换机、路由器、蜂窝电话、个人数字助理等。存储设备102包括接口103、控制部件104、一个或多个NVM(非易失存储器,Non-Volatile Memory)芯片105以及可选地固件存储器110。接口103可适配于通过例如SATA、IDE、USB、PCIE、NVMe、SAS、以太网、光纤通道等方式与主机交换数据。控制部件104用于控制在接口103、NVM芯片105以及固件存储器110之间的数据传输,还用于存储管理、主机逻辑地址到闪存物理地址映射、擦除均衡、坏块管理等。可通过软件、硬件、固件或其组合的多种方式实现控制部件104。控制部件104可以是FPGA(Field-programmable gate array,现场可编程门阵列)、ASIC(Application Specific Integrated Circuit,应用专用集成电路)或者其组合的形式。控制部件104也可以包括处理器或者控制器。控制部件104在运行时从固件存储器110加载固件。固件存储器110可以是NOR闪存、ROM、EEPROM,也可以是NVM芯片105的部分。

[0003] 控制部件104包括闪存接口控制器(或称为介质接口控制器、闪存通道控制器),闪存接口控制器耦合到NVM芯片105,并以遵循NVM芯片105的接口协议的方式向NVM芯片105发出命令,以操作NVM芯片105,并接收从NVM芯片105输出的命令执行结果。已知的NVM芯片接口协议包括“Toggle”、“ONFI”等。

[0004] 存储器目标(Target)是NAND闪存封装内的共享芯片使能(CE,Chip Enable)信号的一个或多个逻辑单元(Logic Unit)。每个逻辑单元具有逻辑单元号(LUN,Logic Unit Number)。NAND闪存封装内可包括一个或多个管芯(Die)。典型地,逻辑单元对应于单一的管芯。逻辑单元可包括多个平面(Plane)。逻辑单元内的多个平面可以并行存取,而NAND闪存芯片内的多个逻辑单元可以彼此独立地执行命令和报告状态。在可从http://www.micron.com/~media/Documents/Products/Other%20Documents/ONFI3_0Gold.ashx获得的“Open NAND Flash Interface Specification(Revision 3.0)”中,提供了关于目标(target)、逻辑单元、LUN、平面(Plane)的含义,其为现有技术的一部分。

[0005] 存储介质上通常按页来存储和读取数据。而按块来擦除数据。块(也称物理块)包含多个页。块包含多个页。存储介质上的页(称为物理页)具有固定的尺寸,例如17664字节。物理页也可以具有其他的尺寸。

[0006] 在公开号为CN1414468A的中国专利申请中,提供了通过执行微指令序列来处理

CPU(Central Processing Unit,中央处理单元)指令的方案。当CPU要处理特定指令时,转换逻辑电路将特定指令转换成与之对应的微指令序列,通过执行微指令序列来实现特定指令的功能。微指令序列或者微指令序列的模板存储在ROM(Read Only Memory,只读存储器)中。在将特定指令转换成微指令序列过程中,可对微指令序列模板进行填充,使之与特定指令相对应。

[0007] 在中国专利申请CN201610009789.6与CN201510253428.1中提供了用于闪存接口控制器的微指令执行方法与装置,中国专利申请CN 201610861793.5提供了微指令序列的调度方法与装置,中国专利申请CN 201611213754.0提供了IO命令处理方法与固态存储设备,中国专利申请CN 201611213755.5提供了大容量NVM接口控制器,将其全文合并于此。闪存接口控制器通常耦合到多个NVM芯片,NVM芯片包括多个LUN(Logic UNit,逻辑单元)或管芯,多个LUN和管芯可以并行响应及访问NVM命令。又由于每个LUN或管芯上可以有多个待处理的NVM命令,因此NVM控制器需要对多个NVM命令的处理过程进行调度,以维护多个在处理中或待处理的NVM命令,或者维护多段用于生成和处理NVM命令的微指令序列的执行。

发明内容

[0008] 存储控制器或其介质接口控制器执行的微指令序列需要被充分测试以交付高质量的产品。微指令序列有针对不同场景的多种可能的被执行方式。为在测试中覆盖多种可能的执行方式,要花费大量时间开发测试用例、执行测试、收集执行结果并分析测试覆盖率,这对测试的快速收敛带来了巨大的挑战。

[0009] 根据本申请的第一方面,提供了根据本申请第一方面的第一存储控制器执行的微指令序列的测试方法,包括:生成访问NVM芯片的一个或多个命令作为测试微指令序列的激励;获取微指令序列的执行轨迹以及同执行轨迹对应的逻辑单元;根据执行轨迹对应的逻辑单元将执行轨迹分组;从各组执行轨迹中提取同所述微指令序列可能的执行路径匹配的 execution path 作为被覆盖的执行路径;累计从每组执行轨迹中提取的多个被覆盖的执行路径,得到所有被覆盖的执行路径,通过所有被覆盖的执行路径与所有可能的执行路径之比,得到执行路径覆盖率。

[0010] 根据本申请第一方面的第一存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第二存储控制器执行的微指令序列的测试方法,还包括:若执行路径覆盖率达到指定阈值,终止所述测试。

[0011] 根据本申请第一方面的第一或第二存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第三存储控制器执行的微指令序列的测试方法,还包括:若执行路径覆盖率未达到指定阈值,根据所有可能的执行路径中尚未被覆盖的执行路径生成第二激励,使得执行被测试的微指令序列以处理同生成的第二激励对应的命令时,执行被测试的微指令序列的所述尚未被覆盖的执行路径。

[0012] 根据本申请第一方面的第一至第三存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第一方面的第四存储控制器执行的微指令序列的测试方法,其中访问不同逻辑单元的作为激励的多个命令,被并发地执行。

[0013] 根据本申请第一方面的第四存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第五存储控制器执行的微指令序列的测试方法,其中所述访问不同逻

辑单元的作为激励的多个命令各自具有不同的参数。

[0014] 根据本申请第一方面的第一至第五存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第一方面的第六存储控制器执行的微指令序列的测试方法,其中随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

[0015] 根据本申请第一方面的第六存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第七存储控制器执行的微指令序列的测试方法,其中通过指定生成的多种类型的命令的每种类型的频度与顺序,和/或应用于所生成的命令的一个或多个约束条件而随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

[0016] 根据本申请第一方面的第一至第七存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第一方面的第八存储控制器执行的微指令序列的测试方法,其中由多个处理器核执行各自的微指令序列以处理激励。

[0017] 根据本申请第一方面的第八存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第九存储控制器执行的微指令序列的测试方法,其中多个处理器核各自执行相同的微指令序列;以及各个处理器核耦合到不同的NVM芯片。

[0018] 根据本申请第一方面的第九存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十存储控制器执行的微指令序列的测试方法,其中每个处理器核运行的微指令序列访问多个逻辑单元。

[0019] 根据本申请第一方面的第八至第十存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第一方面的第十一存储控制器执行的微指令序列的测试方法,其中向各个处理器核施加不同的激励。

[0020] 根据本申请第一方面的第十一存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十二存储控制器执行的微指令序列的测试方法,其中向各个处理器核所施加的激励是具有不同参数的相同类型访问NVM芯片的命令。

[0021] 根据本申请第一方面的第八至第十二存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第十方面的第十三存储控制器执行的微指令序列的测试方法,其中从所述多个处理器核的每个获取微指令序列的执行轨迹以及同执行轨迹对应的逻辑单元。

[0022] 根据本申请第一方面的第一至第十三存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第十方面的第十四存储控制器执行的微指令序列的测试方法,还包括:分析所述微指令序列生成所述微指令序列所有可能的执行路径。

[0023] 根据本申请第一方面的第十四存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十五存储控制器执行的微指令序列的测试方法,其中根据所述微指令序列生成有向图,其中所述微指令序列中的分支指令所在地址、跳转指令所在地址或分支/跳转的目标地址的是所生成的有向图的节点,节点之间的有向边指示了节点所对应的地址所对应的微指令的执行顺序。

[0024] 根据本申请第一方面的第十五存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十六存储控制器执行的微指令序列的测试方法,还包括:合并有向图中的两个或多个节点。

[0025] 根据本申请第一方面的第十五或第十六存储控制器执行的微指令序列的测试方

法,提供了根据本申请第一方面的第十七存储控制器执行的微指令序列的测试方法,还包括:消除有向图中的环。

[0026] 根据本申请第一方面的第十七存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十八存储控制器执行的微指令序列的测试方法,其中若第一节点与第二节点间存在两条有向边,且第一节点与第二节点互为出点与入点,则消除第二节点到第一节点的有向边,以消除环,其中第二节点对应的地址在第一节点对应的地址之后被执行。

[0027] 根据本申请第一方面的第十七或第十八存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第十九存储控制器执行的微指令序列的测试方法,其中通过遍历生成的有向图,当有向图的节点被第二次访问时,取消引起该第二次访问的边,以消除环。

[0028] 根据本申请第一方面的第十五至第十九存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第十方面的第二十存储控制器执行的微指令序列的测试方法,还包括:遍历有向图,获取从有向图的根节点到叶节点的所有可能的执行路径,作为所述微指令序列所有可能的执行路径。

[0029] 根据本申请第一方面的第十五至第二十存储控制器执行的微指令序列的测试方法之一,提供了根据本申请第十方面的第二十一存储控制器执行的微指令序列的测试方法,其中,从执行轨迹中获取同有向图的根节点对应的地址相同的第一存储器地址,从执行轨迹中获取第一存储器地址后续的第二存储器地址,从所有可能的执行路径中搜索同有向图的根节点对应的下一节点对应的地址同第二存储器地址相同的一个或多个执行路径,作为候选执行路径,直到从执行轨迹获取的后续存储器地址,同候选执行路径之一的对应于叶节点的地址相同,以得到匹配的执行路径。

[0030] 根据本申请第一方面的第二十一存储控制器执行的微指令序列的测试方法,提供了根据本申请第一方面的第二十二存储控制器执行的微指令序列的测试方法,其中若从执行轨迹获取的后续存储器地址同任何候选执行路径中相对于当前节点的下一节点对应的地址都不同,从执行轨迹获取下一后续存储器地址同候选执行路径中的下一节点对应的地址做比较,直到获取的下一后续存储器地址同候选执行路径中的下一节点对应的地址相同。

[0031] 根据本申请的第二方面,提供了根据本申请第二方面的第一微指令序列的测试方法,包括:生成测试微指令序列的激励;获取微指令序列的执行轨迹;从执行轨迹中提取同所述微指令序列可能的执行路径匹配的执行路径作为被覆盖的执行路径;累计从执行轨迹中提取的多个被覆盖的执行路径,得到执行路径覆盖率。

[0032] 提供了根据本申请第三方面,提供了根据本申请第三方面的第一微指令序列的测试系统,包括:激励生成模块,用于生成访问NVM芯片的一个或多个命令作为测试微指令序列的激励;激励施加模块,获取激励生成模块产生的激励,并施加给被测试的存储控制器;覆盖率分析模块,从所述存储控制器获取所述微指令序列的执行轨迹,从执行轨迹中提取同所述微指令序列可能的执行路径匹配的执行路径作为被覆盖的执行路径,并累计从执行轨迹中提取的多个被覆盖的执行路径,得到执行路径覆盖率。

[0033] 根据本申请第三方面的第一微指令序列测试系统,提供了根据本申请第三方面的

第二微指令序列的测试系统,还包括:评估单元,从所述存储控制器获取激励的执行结果,从所述激励生成模块获取激励,并比较执行结果与激励的预期结果。

[0034] 根据本申请第三方面的第一或第二微指令序列测试系统,提供了根据本申请第三方面的第三微指令序列的测试系统,其中所述覆盖率分析模块,还获取同执行轨迹对应的逻辑单元,根据执行轨迹对应的逻辑单元将执行轨迹分组;从各组执行轨迹中提取同所述微指令序列可能的执行路径匹配的执行路径作为被覆盖的执行路径。

[0035] 根据本申请第三方面的第一至第三微指令序列测试系统之一,提供了根据本申请第三方面的第四微指令序列的测试系统,其中所述覆盖率测试模块还记录所述微指令序列所有可能的执行路径。

[0036] 根据本申请第三方面的第四微指令序列测试系统,提供了根据本申请第三方面的第五微指令序列的测试系统,其中所述所有可能的执行路径是分析所述微指令序列得到的。

[0037] 根据本申请第三方面的第一至第五微指令序列测试系统之一,提供了根据本申请第三方面的第六微指令序列的测试系统,其中所述覆盖率分析模块识别执行路径覆盖率达到指定阈值,指示所述测试系统终止测试。

[0038] 根据本申请第三方面的第一至第六微指令序列测试系统之一,提供了根据本申请第三方面的第七微指令序列的测试系统,还包括:所述覆盖率分析模块识别执行路径覆盖率未达到指定阈值,指示所述激励生成模块根据所有可能的执行路径中尚未被覆盖的执行路径生成第二激励,使得执行被测试的微指令序列以处理同生成的第二激励对应的命令时,执行被测试的微指令序列的所述尚未被覆盖的执行路径。

[0039] 根据本申请第三方面的第一至第七微指令序列测试系统之一,提供了根据本申请第三方面的第八微指令序列的测试系统,其中所述存储控制器并发地执行访问不同逻辑单元的作为激励的多个命令。

[0040] 根据本申请第三方面的第八微指令序列测试系统,提供了根据本申请第三方面的第九微指令序列的测试系统,其中所述访问不同逻辑单元的作为激励的多个命令各自具有不同的参数。

[0041] 根据本申请第三方面的第一至第九微指令序列测试系统之一,提供了根据本申请第三方面的第十微指令序列的测试系统,其中所述激励生成模块随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

[0042] 根据本申请第三方面的第十微指令序列测试系统,提供了根据本申请第三方面的第十一微指令序列的测试系统,其中所述激励生成模块通过指定生成的多种类型的命令的每种类型的频度与顺序,和/或应用于所生成的命令的一个或多个约束条件而随机地生成访问NVM芯片的多个命令作为测试微指令序列的激励。

[0043] 根据本申请第三方面的第一至第十一微指令序列测试系统之一,提供了根据本申请第三方面的第十二微指令序列的测试系统,其中所述存储控制器的多个处理器核执行各自的微指令序列以处理激励。

[0044] 根据本申请第三方面的第十二微指令序列测试系统,提供了根据本申请第三方面的第十三微指令序列的测试系统,其中所述存储控制器的多个处理器核各自执行相同的微指令序列;以及各个处理器核耦合到不同的NVM芯片。

[0045] 根据本申请第三方面的第十三微指令序列测试系统,提供了根据本申请第三方面的第十四微指令序列的测试系统,其中每个处理器核运行的微指令序列访问多个逻辑单元。

[0046] 根据本申请第三方面的第十二至第十四微指令序列测试系统之一供了根据本申请第三方面的第十五指令序列的测试系统,其中激励施加模块向各个处理器核施加不同的激励。

[0047] 根据本申请第三方面的第十五指令序列测试系统,提供了根据本申请第三方面的第十六指令序列的测试系统,其中向各个处理器核所施加的激励是具有不同参数的相同类型访问NVM芯片的命令。

[0048] 根据本申请第三方面的第十二至第十六指令序列测试系统之一供了根据本申请第三方面的第十七指令序列的测试系统,中所述覆盖率分析模块从所述多个处理器核的每个获取微指令序列的执行轨迹以及同执行轨迹对应的逻辑单元。

附图说明

[0049] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0050] 图1展示了现有技术的存储设备的框图;

[0051] 图2A是根据本申请实施例的控制部件的介质接口控制器的框图;

[0052] 图2B是根据本发明又一实施例的固态硬盘控制器的NVM接口控制器的框图

[0053] 图3是根据本申请实施例的从微指令序列生成有向图节点的示意图;

[0054] 图4,展示了从微指令序列生成的有向图的部分;

[0055] 图5A展示根据本申请实施例的从微指令序列得到的有向图;

[0056] 图5B展示了遍历图5A得到的所有4条可能的执行路径;

[0057] 图6展示了根据本申请实施例的测试系统;

[0058] 图7展示了根据本申请实施例的所获取的微指令序列的执行轨迹的示意图;

[0059] 图8展示了微指令序列的执行轨迹与执行路径匹配的示意图;

[0060] 图9是根据本申请实施例的覆盖率测试方法的示意图;以及

[0061] 图10是根据本申请实施例的激励生成模块的示意图;以及

[0062] 图11是根据本申请又一实施例的覆盖率测试方法的示意图。

具体实施方式

[0063] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0064] 图2A是根据本申请实施例的控制部件的介质接口控制器的框图。图2A中的介质接口控制器包括消息队列210与NVM命令处理单元220。图2A的实施例中,消息队列210用于接

收来自控制部件(也参看图1的控制部件)的访问NVM芯片的消息。来自控制部件的消息可包括指示读出、写入、删除NVM芯片的消息,还可以包括指示读取NVM芯片状态、读取或设置NVM芯片特征(Feature)的消息,以及也可以包括用户自定义消息。NVM命令处理单元220从消息队列210获取消息,并根据消息的指示向NVM芯片发送符合NVM芯片接口标准的NVM接口命令或依照NVM芯片接口标准从NVM接收数据或状态。作为举例,根据本申请实施例的NVM命令处理单元,通过例如执行指令或微指令,对NVM接口命令实施调度。消息队列210包括一个或多个消息队列。

[0065] NVM命令处理单元220包括例如2个处理器核(核1与核2),通过处理器核耦合到多个NVM芯片。在图2A的实施例中,NVM命令处理单元220通过2个通道(CH1与CH2)耦合到4颗NVM芯片,核1耦合第一通道,核2耦合第二通道。每个NVM芯片包括2个LUN。在通道CH 1的NVM芯片(NVM 0与NVM 1)各自提供LUN 0与LUN 1,在通道CH 2的NVM芯片(NVM 2与NVM 3)各自提供其LUN0与LUN1。可以理解的,NVM接口控制器可耦合更多的通道,并访问更多的NVM芯片与更多的LUN。

[0066] 介质接口控制器能够对来自多个消息队列的多个访问NVM芯片的消息并行处理。NVM命令处理单元可包括多个处理器核,每个处理器核可运行多个线程,每个处理器核或线程,并行处理多个NVM接口命令。处理器核或执行微指令序列的线程访问各自对应的NVM芯片。将可被执行的微指令序列被称作线程。由于同一微指令序列在每次执行时拥有自己的执行状态,从而可基于同一微指令序列创建多个线程。在NVM命令处理单元220中还为每个线程存储执行状态。根据本申请的实施例,基于所要访问的LUN来创建或使用线程。例如使用线程1来访问LUN1,和/或使用线程2来访问LUN2。

[0067] 图2B是根据本发明又一实施例的固态硬盘控制器的NVM接口控制器的框图。图2B中的NVM接口控制器包括命令队列201与NVM命令处理单元202。NVM处理单元202包括多个处理器核(示出了核1与核2),各个处理器核通过执行微指令序列来访问对应的NVM芯片。为微指令序列的执行分配上下文资源。作为举例,上下文资源同LUN对应。例如,处理器核1负责访问LUN 0、LUN 1、LUN 2与LUN 3。为访问LUN0而分配上下文1给核1,为访问LUN 3而分配上下文2核1。核1执行的微指令序列使用上下文1访问LUN 1而使用上下文2访问LUN 3。图2B使用了由核1使用的两个上下文。可选地,为核1所管理的每个LUN提供对应的上下文资源。上下文记录了与LUN相关的微执行序列执行的状态,例如,数据缓存、微指令序列的程序计数器、定时器等。类似地,图2B的实施例中,核2执行的微指令序列使用上下文3访问LUN 5而使用上下文4访问LUN 7。

[0068] 根据本申请的实施例,分析微指令序列,以提取微指令序列的所有可能的执行路径。并在测试过程中,捕获处理器核的微指令的执行规则,以识别哪些可能的执行路径被执行,并合同所有被执行的路径,以得到微指令序列的代码覆盖率。

[0069] 图3是根据本申请实施例的从微指令序列生成有向图节点的示意图。

[0070] 参看图3,微指令序列310包括多条微指令,每条微指令可位于指定的存储器地址(由图3的“地址”字段指示),微指令具有指定的微指令代码,处理器核依据微指令代码识别指定微指令,并执行微指令所指定的操作。图3中还展示了每条微指令对应的汇编代码,以利于所属领域技术人员理解微指令的含义。

[0071] 通常,处理器核按存储器地址的顺序(例如,图3中向下,或地址增序),依次执行每

条微指令。一些微指令是分支/跳转微指令,当执行分支/跳转微指令时,微指令执行的顺序发生变化,处理器核接下来执行由分支/跳转微指令所指示的存储器地址处的微指令。为获得微指令序列的所有可能的执行路径,根据本申请的实施例,将微指令序列转换为有向图。有向图包括点与边,点指示一条微指令,或微存储器所在的存储器地址。有向图的边具有方向,从出点指向入点,并且意味着在执行完出点所对应的微指令后,将执行入点所执行的微指令。可选地,在出点与入点对应的微指令之间,在微指令序列可能存在一条或多条其他微指令,这些微指令将被必然执行,而不影响对微指令执行路径的覆盖率的计算。

[0072] 根据本申请的实施例,从微指令序列310生成有向图的节点集合320。图3中,有向图的节点集合被展示为表格,表格的每个条目展示了微指令对应的存储器地址与其汇编代码。可以理解地,展示汇编代码是为了理解的目的,有向图的节点集合320的每个元素可以仅记录微指令对应的存储器地址。

[0073] 为生成有向图的节点集合320,选取微指令序列310中的分支微指令、跳转微指令与具有分支/跳转含义的退出(exit)、让步(Yield)等微指令。可选地,还选取微指令序列310中的分支微指令/跳转微指令的目标地址,添加到节点集合320。目标地址作为节点集合320的元素,不是必须的。处理器核在目标地址处继续向下执行,直到执行到下一作为有向图节点的分支/跳转微指令。

[0074] 根据有向图节点集合320,生成有向图。节点集合320中的每个元素,被作为有向图的节点。依据节点集合320中每个元素的所代表的分支/跳转的一个或两个目标地址,建立从有向图的一个节点,指向另一个或两个节点的有向边。

[0075] 以及可选地,所得到的有向图中的两个或多个节点可被合并。例如,两个节点间仅存在一条有向边,并且该有向边的出点不作为任何其他有向边的出点,则这两个节点可以合并为一个节点。作为举例,被顺序执行的两个节点将被合并。

[0076] 由于循环结构的存在,从微指令序列生成的有向图可能存在环。图4展示了根据本申请实施例的有向图消除环的示意图。

[0077] 参看图4,展示了从微指令序列生成的有向图的部分,包括由地址“0x00000049”、“0x0000004b”、“0x0000004e”与“0x00000052”所指示的有向图节点,及其边。其中节点“0x0000004b”与“0x0000004e”形成循环结构,存在有向边从节点“0x0000004b”指向“0x0000004e”(410),而同时也存在有向边从节点“0x0000004e”指向“0x0000004b”(420)。对此结构,删除从节点“0x0000004e”指向“0x0000004b”的有向边,以消除有向图中的环。一般地,遍历有向图,当有向图的节点(例如,节点“0x0000004b”)被第二次访问时,取消引起该第二次访问的指向该节点(例如,节点“0x0000004b”)的边(420),以消除环。

[0078] 图5A展示根据本申请实施例的从微指令序列得到的有向图。

[0079] 图5A展示的有向图的节点由微指令在存储器中的地址所指示,并且有向图已被消除了环。图5A的有向图的节点中,以“0x00000000(0)”为例,括号“()”前的字符指示节点对应的微指令在存储器中的地址,而由括号内的数字指示节点的序号。有向图的边指示当前执行边的出点时,接下来将执行到的入点。从有向图的根节点到叶节点的任意路径,代表了微指令序列的一种可能的执行路径。

[0080] 遍历由图5A所示的有向图,得到从有向图的根节点到叶节点的所有可能路径,并在图5B中展示。

[0081] 图5B展示了遍历图5A得到的所有4条可能的执行路径,每条路径以根节点“0x00000000(0)”开始,以有向图的叶节点为结束。

[0082] 图6展示了根据本申请实施例的测试系统。

[0083] 测试系统用于测试NVM命令处理单元610(也参看图2A或图2B的NVM命令处理单元)。测试系统的激励生成模块,用于生成例如访问NVM芯片的一个或多个命令,以及可选地用于命令的参数和/或多个命令之间的顺序等约束条件,作为用于要施加给NVM命令处理单元的激励。激励施加模块630接收激励生成单元620提供的激励,转换为NVM命令处理单元610可接收的格式与接口形式,并提供给NVM命令处理单元610。执行结果获取单元提取NVM命令处理单元610的执行结果,并提供给评估单元660。评估单元660接收执行结果获取单元640提供的执行结果,将执行结果与激励生成单元620提供的与激励对应的预期结果做比较,以评估NVM命令处理单元610响应所生成的激励的正确性。

[0084] 覆盖率分析单元650获取NVM命令处理单元610执行微指令序列的执行轨迹,并通过执行轨迹,识别对微指令序列的测试的路径覆盖率。覆盖率分析单元650获取的执行轨迹,是例如按被执行顺序的每条微指令的存储器地址序列。可选地,连同被执行的每条微指令的存储器地址,一同获取执行该微指令的处理器核、上下文编号和/或执行微指令所访问的逻辑单元(LUN)。

[0085] 图7展示了根据本申请实施例的所获取的微指令序列的执行轨迹的示意图。

[0086] 图7展示的执行轨迹,是按时间先后顺序从NVM命令处理单元610获取的执行轨迹。由数字标识的方格指示微指令的存储器地址。其中,越左侧的方格对应的存储器地址被接收的时间越早,越右侧的方格对应的存储器地址被接收的时间越晚。方格中的数字指示例如微指令被执行时所访问的逻辑单元(LUN)。

[0087] 图7展示的执行轨迹,混杂了访问多个逻辑单元(LUN)的微指令的存储器地址。依据执行轨迹对应的逻辑单元(LUN),对执行轨迹分组,每组的执行轨迹对应的微指令访问相同的逻辑单元。图7展示了访问逻辑单元(LUN 0)的执行轨迹,访问逻辑单元(LUN 1)的执行轨迹,与访问逻辑单元(LUN 2)的执行轨迹。每组执行轨迹保留执行轨迹被获取时的顺序。

[0088] 可选地,所获取的执行轨迹仅访问同一逻辑单元,因而属于同一组。

[0089] 返回参看图6,覆盖率分析单元650,将访问例如逻辑单元0的执行轨迹,同从微指令序列得到的可能的执行路径(参看图5A与图5B)进行匹配,以识别微指令序列可能的执行路径中的哪个或哪些已经被NVM处理单元执行,从而得到微指令序列的路径覆盖率。

[0090] 例如,图5A与图5B中,微指令序列的执行路径均以地址“0x00000000”开始。从访问例如逻辑单元0的执行轨迹中找到存储器地址“0x00000000”,并将其后的执行轨迹同图5A与图5B中的微指令序列的执行路径进行匹配。

[0091] 图8展示了微指令序列的执行轨迹与执行路径匹配的示意图。

[0092] 图8中,左侧展示了从NVM命令处理单元获取的执行轨迹片段,从存储器地址“0x00000000”开始,到存储器地址“0x0000034e”结束。而图8的右侧展示了微指令序列的所有可能执行路径。从执行轨迹中,寻找执行路径的路口(例如“0x00000000”),以此作为匹配的起点。接下来的执行轨迹是存储器地址“0x00000001”,从所有可能的执行路径中寻找第二节点是存储器地址“0x00000001”的执行路径,作为候选执行路径,而其他执行路径(例如第二节点是“0x00000003”的执行路径)可被排除在搜索范围之外。接下来,执行轨迹轨迹的

存储器地址“0x00000268”，从所有剩余的可能执行路径中寻找第三节点是存储器地址“0x00000268”的执行路径，作为候选执行路径。以此方式，直到完整匹配了候选执行路径之一，将该被完整匹配的执行路径标记为已被覆盖。

[0093] 可选地，微指令序列有多个入口。对应地，根据微指令序列的每个入口生成对应于微指令序列的有向图，并遍历各个有向图得到微指令序列的所有可能执行路径。以及在匹配执行轨迹与执行路径时，以微指令序列的多个入口任何一个作为匹配的起始。

[0094] 可选地，由于有向图的节点被合并，或者有向图的节点仅包括分支/跳转指令及其目标地址，一些微指令不在图5A与图5B展示的执行路径中。在执行轨迹与执行路径的匹配过程中，在任何执行路径中都无法匹配的来自执行轨迹的存储器地址被丢弃。

[0095] 以及还统计微指令序列所有可能的执行路径的哪些已经被捕获的执行轨迹匹配，通过已经被捕获的执行轨迹匹配的执行路径与所有可能的执行路径之比，得到微指令序列的执行路径覆盖率。

[0096] 图9是根据本申请实施例的覆盖率测试方法的示意图。

[0097] 作为举例，图9的实施例对NVM命令处理单元的4个处理器核（分别记为核1、核2、核3与核4）进行覆盖率测试。为处理器核的每个生成激励（910），激励例如是待由处理器核处理的访问NVM芯片的一个或多个命令。将生成的激励施加给处理器核。作为举例，生成一份激励，而应用于每个处理器核，从而每个处理器核执行相同的微指令序列，处理相同的作为激励的命令，并且访问不同的逻辑单元。

[0098] 根据处理器核所执行的微指令序列生成其所有可能的执行路径。例如，根据图3到图5所示的方式从微指令序列生成其所有可能的执行路径。

[0099] 选择处理器核之一，并从选择的处理器核提取微指令序列的执行轨迹（920）。获取的执行轨迹是处理器核执行微指令序列时，按执行顺序的微指令地址序列，以及同每个微指令地址相关联的执行该微指令所访问的逻辑单元标识。进一步地，按逻辑单元标识，对提取的微指令序列的执行轨迹分组。

[0100] 从访问相同逻辑单元的微指令序列的执行轨迹中提取同可能的执行路径匹配的执行路径，作为被覆盖的执行路径（930）。

[0101] 对于从访问多个逻辑单元的每个的微指令序列的执行轨迹得到的多个被覆盖路径求和，得到所有被覆盖的执行路径，并依据所有被覆盖的执行路径占所有可能的执行路径之比，得到路径覆盖率（940）。

[0102] 可选地，还对所有被覆盖的执行路径和/或所有可能的执行路径中未被覆盖的执行路径进行分析（950），根据分析结果，指导用于覆盖率测试的激励的生成（910）。例如，根据分析结果，识别出对应于第一类型的访问NVM芯片的命令的执行路径未被覆盖，则在接下来生成的激励中，提高第一类型的NVM芯片的命令的比例。可选地或进一步地，将路径覆盖率达到指定阈值作为测试停止的标准。

[0103] 图10是根据本申请实施例的激励生成模块的示意图。

[0104] 激励生成模块用于生成例如访问NVM芯片的一个或多个命令。访问NVM芯片的命令具有多种类型，例如，读命令、写命令、擦除命令、参数设置命令与复位命令。激励生成模块根据根据每种命令类型，生成同命令类型对应的命令。

[0105] 激励生成模块包括读命令测试模块、写命令测试模块、擦除命令测试模块、参数设

置命令模块与复位命令测试模块。读命令测试模块用于生成读命令，写命令测试模块用于生成写命令，擦除命令测试模块用于生成擦除命令，参数设置命令测试模块用于生成参数设置命令，而复位命令测试模块用于生成复位命令。读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块与复位命令测试模块都耦合到激励选择器，激励选择器选择读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块与复位命令测试模块所生成的一个或多个命令，作为激励生成模块的输出。

[0106] 可选地，激励选择器还维护从读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块与复位命令测试模块获取命令的顺序和/或权重(或频度)，以向生成的激励施加约束条件。例如，在发出对指定地址的读命令之前，要先向该指定地址施加写命令；在向指定地址施加写命令之前，要使该指定地址所在的存储单元被擦除。

[0107] 访问NVM芯片的命令具有指定的格式。例如，读命令可被设置读取的地址、读取的数据的长度、读取的类型(读页缓存、读物理页、读重做等)。从读命令被发出到查询读命令的结果的时间间隔也可以被设置。还可设置两条读命令的时间间隔，以及待生成的读命令的数量。这些可设置项被作为多种约束条件提供读命令测试模块。读命令测试模块根据一种或多种约束条件，生成读命令作为激励。

[0108] 类似地，写命令可被设置写入的地址、写入的数据的长度、写入的类型(为延迟优化的写命令、为寿命优化的写命令等)。从写命令被发出到查询写命令的结果的时间间隔也可以被设置。还可设置两条相邻写命令的时间间隔，以及待生成的写命令的数量。这些可设置项被作为多种约束条件提供写命令测试模块。写命令测试模块根据一种或多种约束条件，生成写命令作为激励。

[0109] 类似地，擦除命令可被设置擦除的地址、擦除的类型(为延迟优化的擦除命令、为寿命优化的擦除命令等)。从擦除命令被发出到查询擦除命令的结果的时间间隔也可以被设置。还可设置两条相邻擦除命令的时间间隔，以及待生成的擦除命令的数量。这些可设置项被作为多种约束条件提供擦除命令测试模块。擦除命令测试模块根据一种或多种约束条件，生成擦除命令作为激励。

[0110] 类似地，参数设置命令可被设置参数的地址、参数的值和/或参数的长度。还可设置两条相邻参数设置命令的时间间隔，参数设置命令的类型(作用范围为逻辑单元，作用范围为管芯，获取参数等)以及待生成的参数设置命令的数量。这些可设置项被作为多种约束条件提供参数设置命令测试模块。参数设置命令测试模块根据一种或多种约束条件，生成参数设置命令作为激励。

[0111] 类似地，可设置产生复位命令的时机，和/或两条相邻复位命令的时间间隔。这些可设置项被作为多种约束条件提供复位命令测试模块。复位命令测试模块根据一种或多种约束条件，生成复位命令作为激励。

[0112] 进一步地，向读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块和/或复位命令测试模块提供使用一种或多种约束条件的频度，和/或一种或多种约束条件的波动。从而产生多样化的命令。例如，生成的多个读命令所访问的地址符合一定的概率分布。

[0113] 图11是根据本申请又一实施例的覆盖率测试方法的示意图。

[0114] 图11的实施例对NVM命令处理单元的4个处理器核(分别记为核1、核2、核3与核4)

进行覆盖率测试。为处理器核的每个生成激励(1110、1112、1114与1116)。将生成的激励施加给处理器核。根据图11的实施例,为每个处理器核生成的激励是不同的,从而使得各个处理器核执行微指令序列的不同路径,以加快覆盖率测试过程。每个处理器核执行相同的微指令序列,处理不同的作为激励的命令,并且访问不同的逻辑单元。响应于处理不同的作为激励的命令,各个处理器核上执行微指令序列的不同路径。

[0115] 在可选的实施方式中,相同的作为激励的命令(例如,读命令或写命令)被施加给各个处理器核,而为每个处理器核所接收的命令施加不同的约束条件(也参看图10)。例如,为读命令所访问的地址施加具有指定概率分布的约束,从而各个处理器核所接收的待处理读命令所访问的地址符合指定的概率分布。

[0116] 根据处理器核所执行的微指令序列生成其所有可能的执行路径。例如,根据图3到图5所示的方式从微指令序列生成其所有可能的执行路径。

[0117] 对于每个处理器核,从处理器核提取微指令序列的执行轨迹(1120、1122、1124与1126)。获取的执行轨迹是处理器核执行微指令序列时,按执行顺序的微指令地址序列,以及同每个微指令地址相关联的执行该微指令所访问的逻辑单元标识。进一步地,按逻辑单元标识,对提取的微指令序列的执行轨迹分组。

[0118] 从访问相同逻辑单元的微指令序列的执行轨迹中提取同可能的执行路径匹配的执行路径,作为被覆盖的执行路径(1130、1132、1134与1136)。

[0119] 对于从访问多个逻辑单元的每个的微指令序列的执行轨迹得到的多个被匹配路径求和,得到所有被覆盖的执行路径,并依据所有被覆盖的执行路径占所有可能的执行路径之比,得到路径覆盖率(1140)。

[0120] 可选地,还对所有被覆盖的执行路径和/或所有可能的执行路径中未被覆盖的执行路径进行分析(1150),根据分析结果,调整用于一个或多个处理器核的覆盖率测试的激励的生成策略(910)。激励的生成策略,包括例如激励选择器从读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块与复位命令测试模块获取命令的顺序和/或权重(或频度),以及读命令测试模块、写命令测试模块、擦除命令测试模块、参数设置命令模块和/或复位命令测试模块提供使用一种或多种约束条件的频度,和/或一种或多种约束条件的波动。依然可选地,根据分析结果,识别出对应于第一类型的访问NVM芯片的命令的执行路径未被覆盖,则在接下来生成的激励中,提高第一类型的NVM芯片的命令的比例。

[0121] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

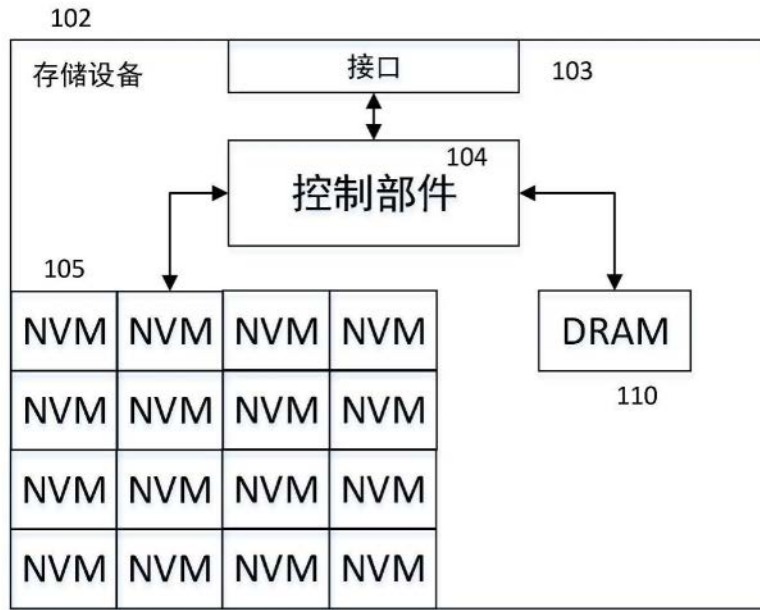


图1

介质接口控制器

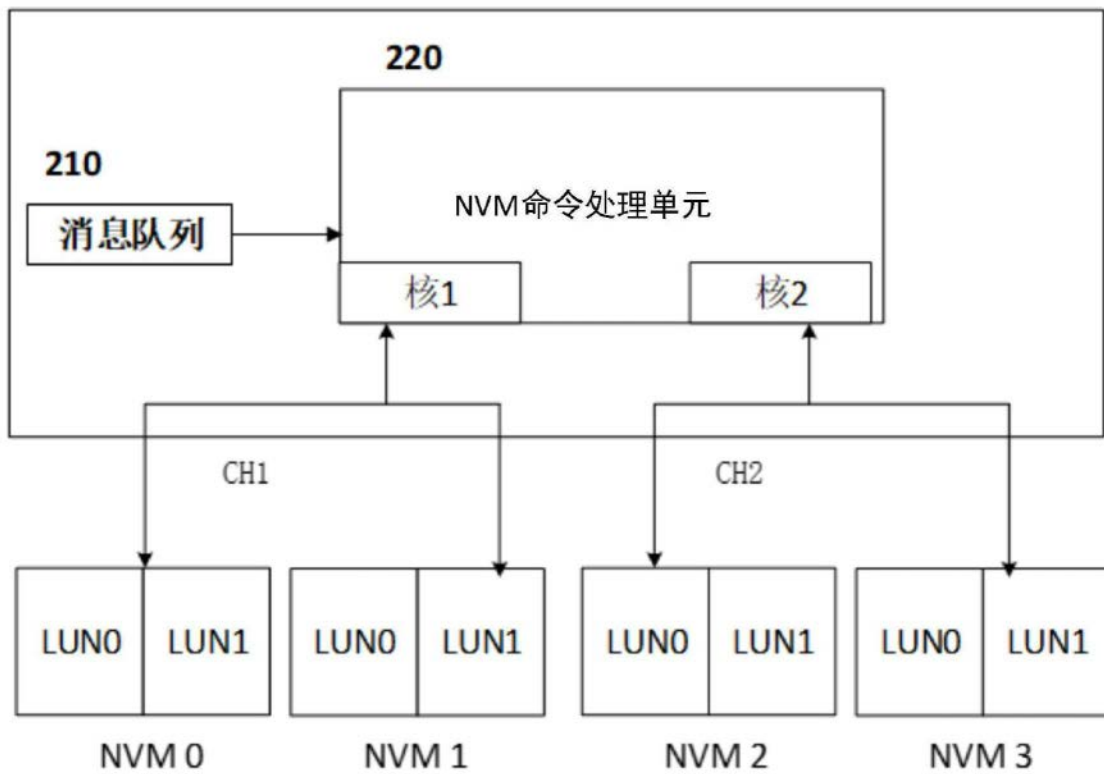


图2A

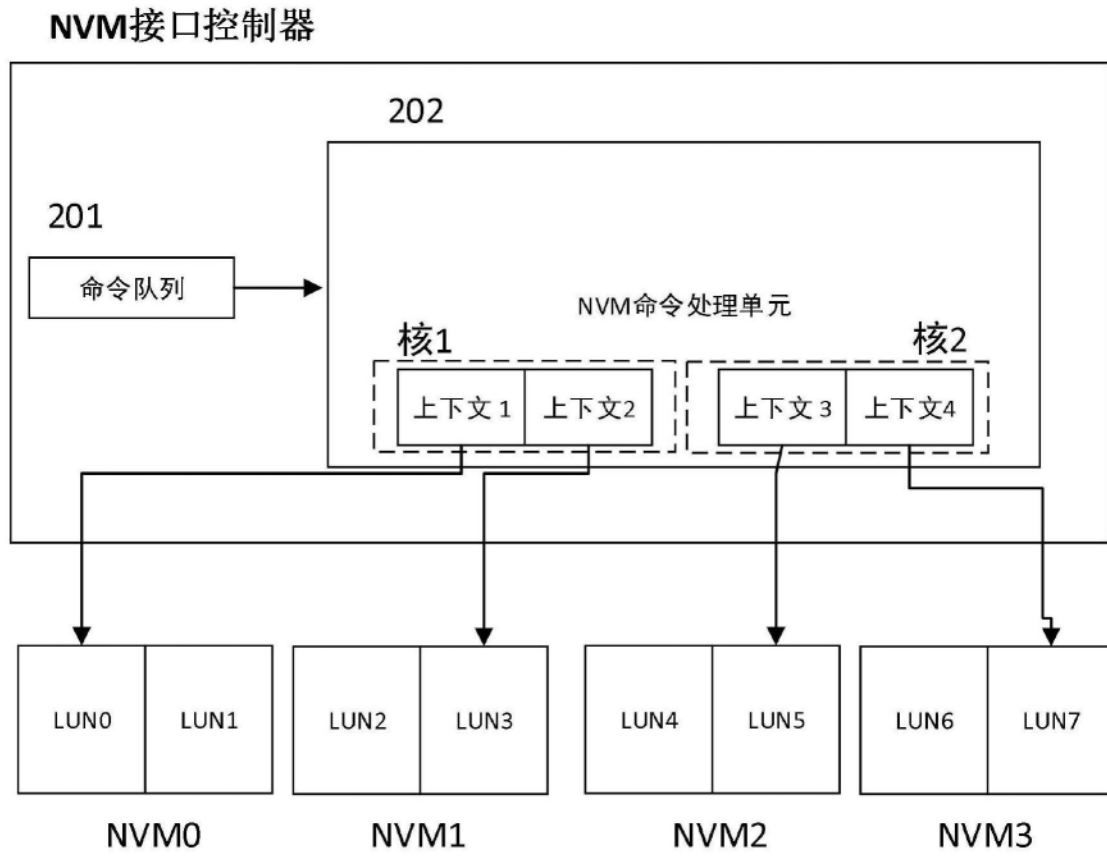


图2B

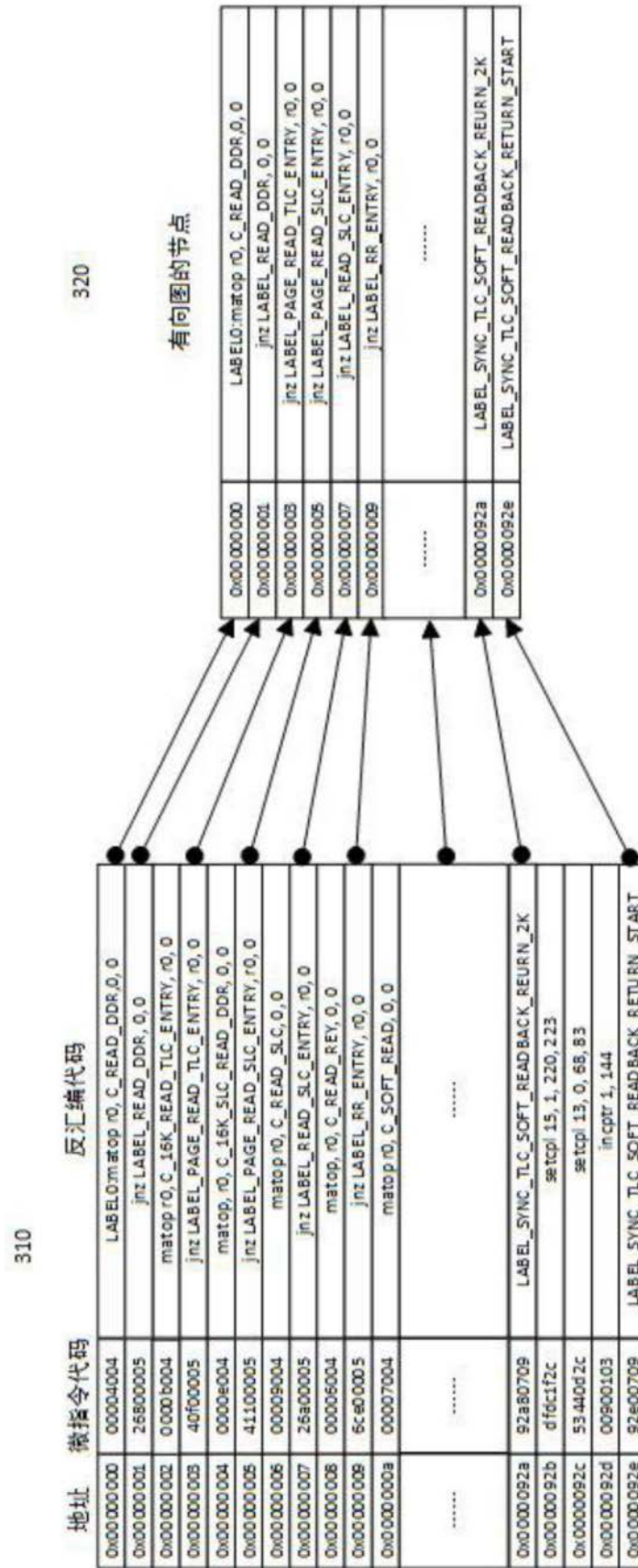


图3

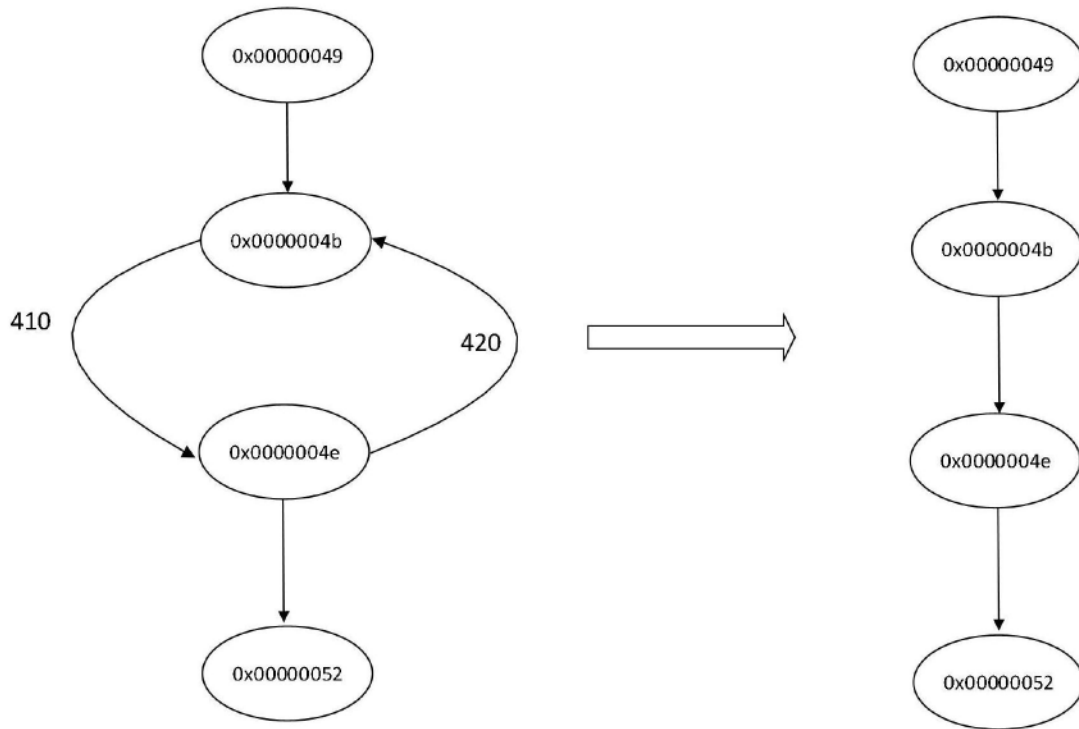


图4

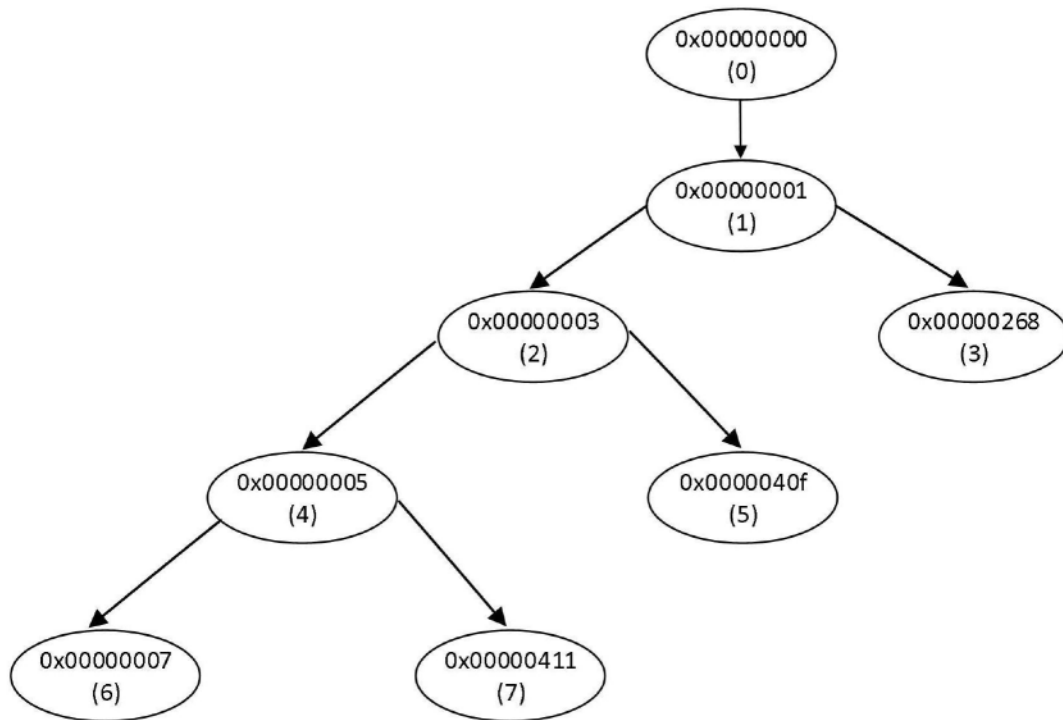


图5A

0x00000000	0x00000001	0x00000268	结束		
0x00000000	0x00000001	0x00000003	0x0000040f	结束	
0x00000000	0x00000001	0x00000003	0x00000005	0x00000411	结束
0x00000000	0x00000001	0x00000003	0x00000005	0x00000007	结束

图5B

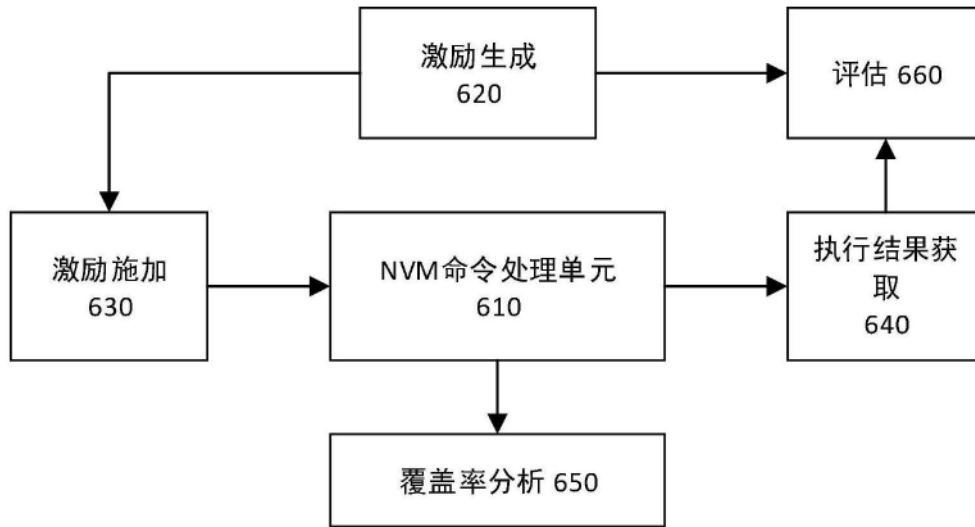


图6

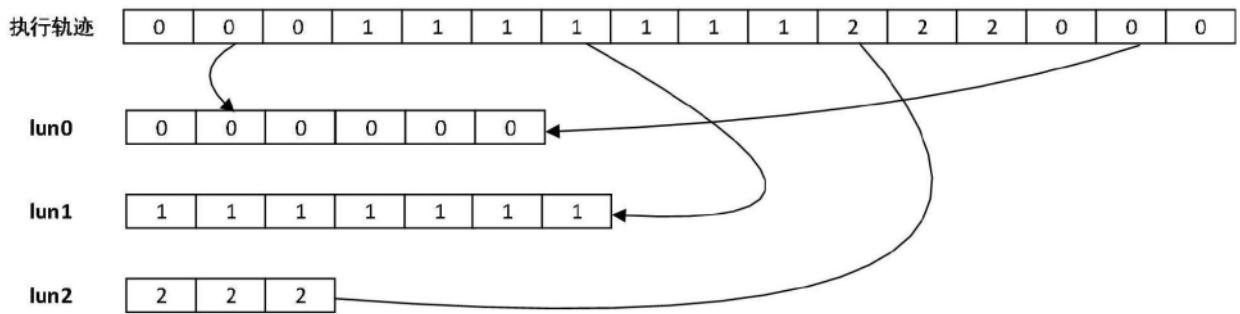


图7

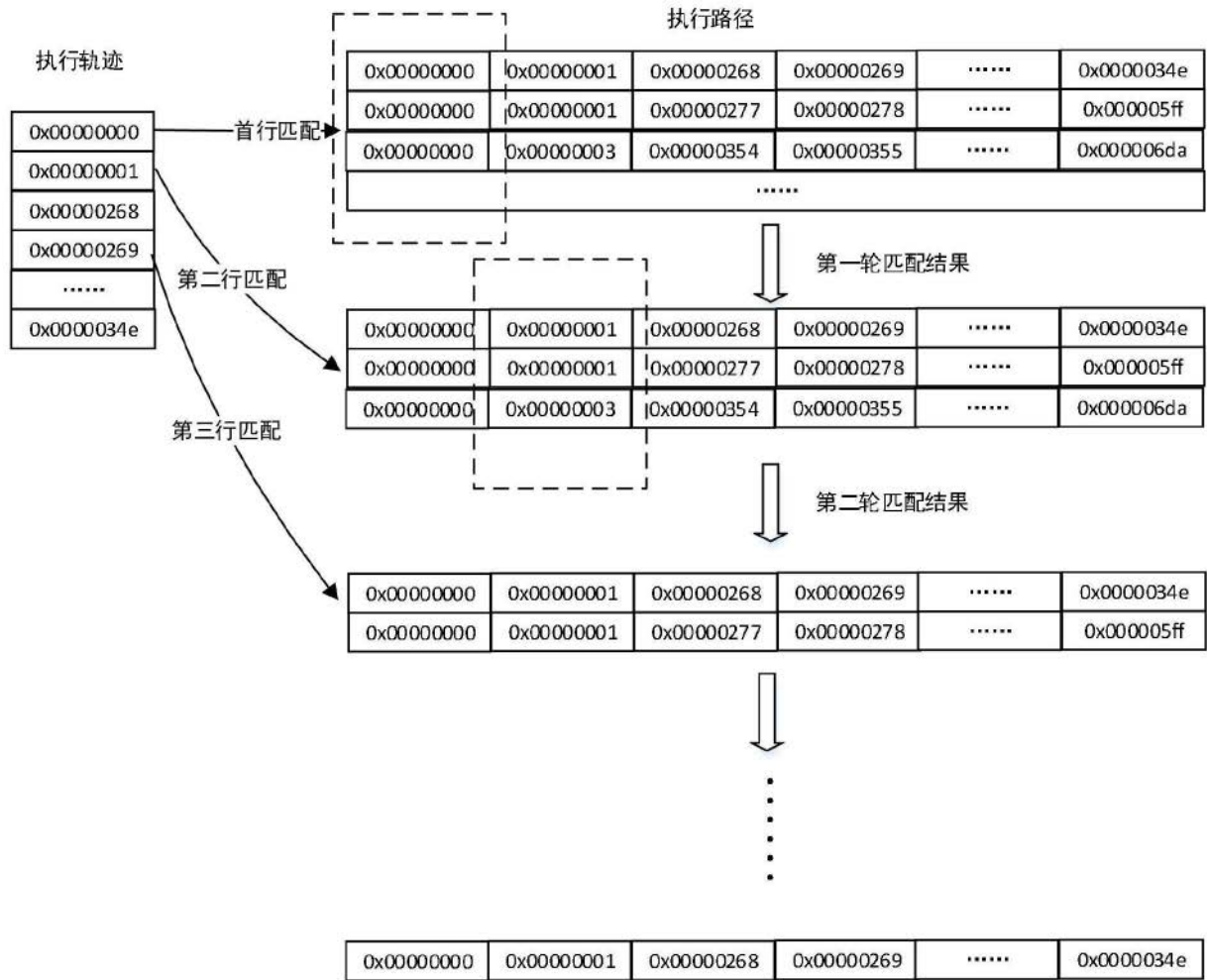


图8

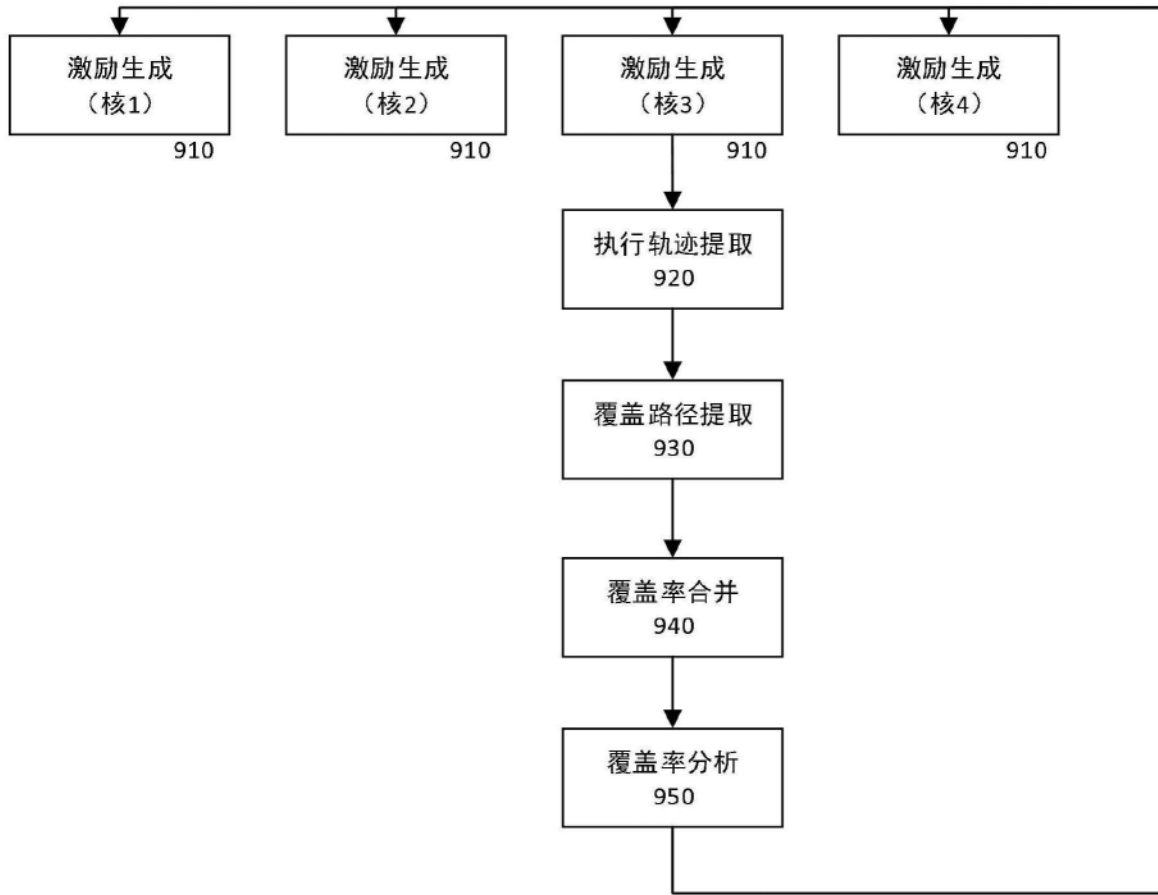


图9

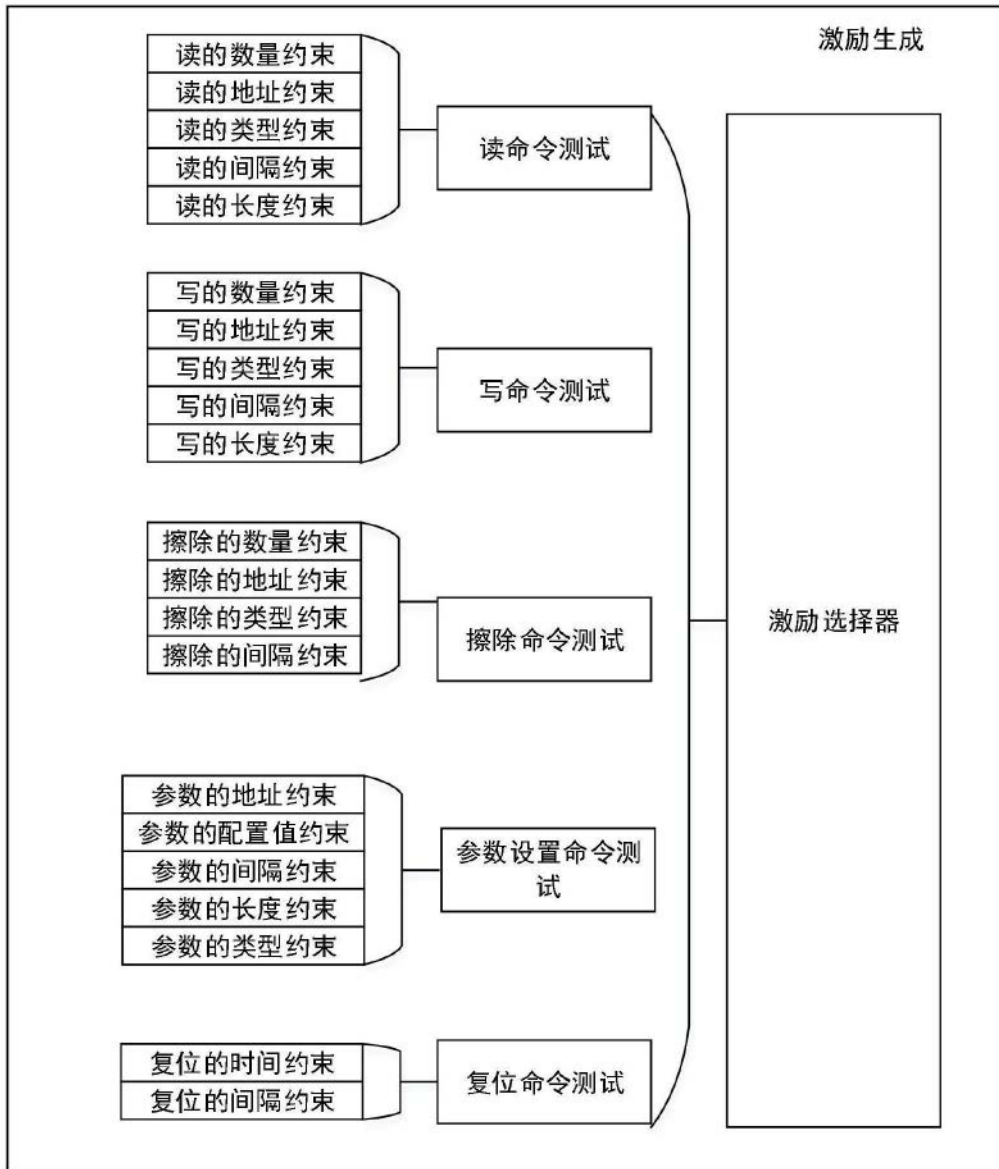


图10

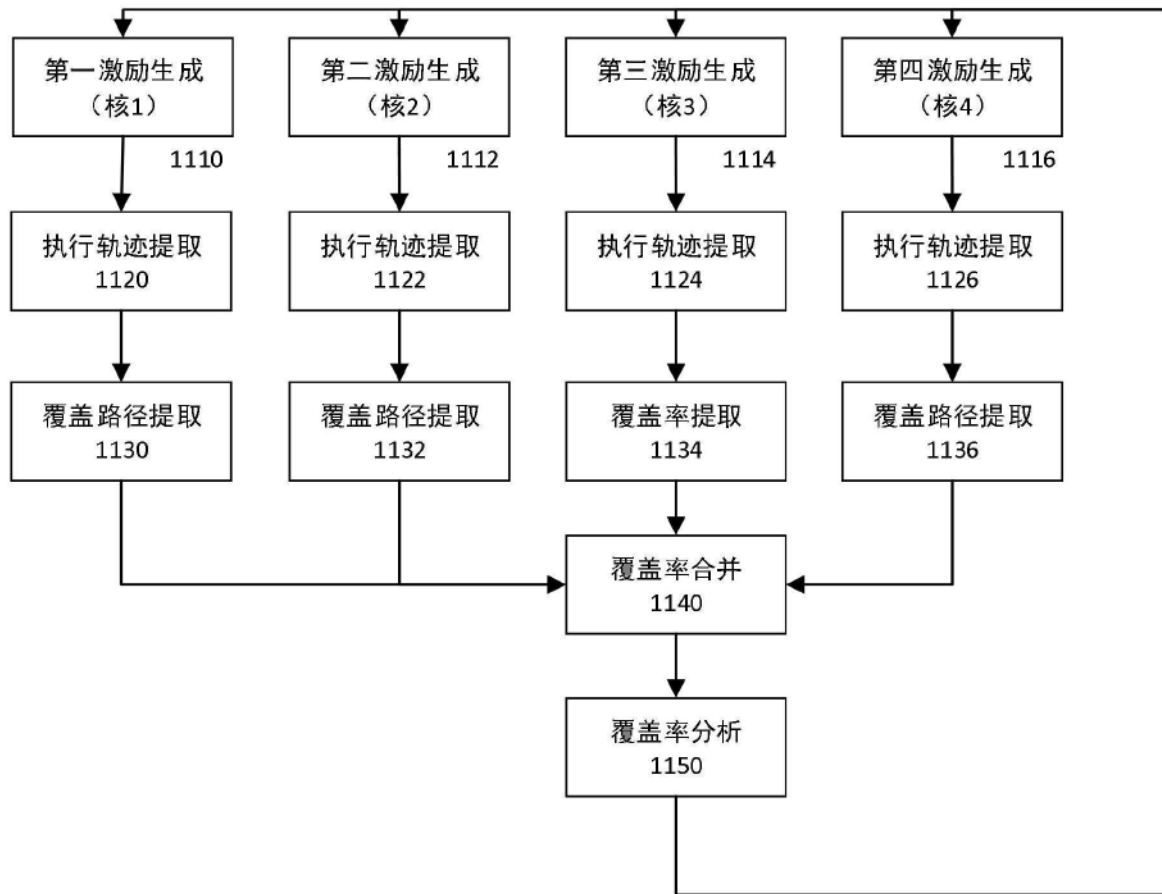


图11