



(12) 发明专利申请

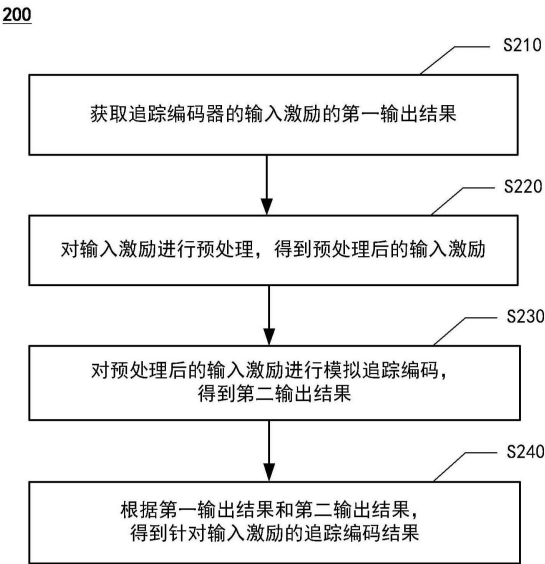
(10) 申请公布号 CN 116166324 A
(43) 申请公布日 2023. 05. 26

(21) 申请号 202211546001.7
(22) 申请日 2022.12.02
(71) 申请人 北京奕斯伟计算技术股份有限公司
地址 101111 北京市大兴区北京经济技术
开发区科创十街18号院3号楼1层101
室
(72) 发明人 惠君龙 姜文
(74) 专利代理机构 中科专利商标代理有限责任
公司 11021
专利代理师 杨静
(51) Int.Cl.
G06F 9/32 (2018.01)
G06F 9/30 (2018.01)

权利要求书3页 说明书12页 附图5页

(54) 发明名称
追踪编码器的验证方法、装置和电子设备

(57) 摘要
本公开提供了一种追踪编码器的验证方法、装置和电子设备,该方法包括:获取追踪编码器的输入激励的第一输出结果;其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果;对输入激励进行预处理,得到预处理后的输入激励;对预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及,根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。



1. 一种追踪编码器的验证方法,其特征在于,包括:

获取追踪编码器的输入激励的第一输出结果;其中,所述第一输出结果为对所述输入激励进行被测追踪编码生成的输出结果;

对所述输入激励进行预处理,得到预处理后的输入激励;

对所述预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及

根据所述第一输出结果和所述第二输出结果,得到针对所述输入激励的追踪编码结果。

2. 根据权利要求1所述的验证方法,其特征在于,所述输入激励包括指令计数器激励和寄存器配置激励;其中,所述对所述输入激励进行预处理,得到预处理后的输入激励,包括:

根据所述指令计数器激励的输入时刻和所述寄存器配置激励的输入时刻,生成与所述指令计数器激励和所述寄存器配置激励分别对应的输入周期。

3. 根据权利要求1所述的验证方法,其特征在于,所述预处理后的输入激励中每个输入激励的输入周期彼此不同。

4. 根据权利要求2所述的验证方法,其特征在于,所述对所述预处理后的输入激励进行模拟追踪编码,得到第二输出结果,包括:

获取预处理后的指令计数器激励中未执行的N个指令计数器激励;其中,第一指令计数器激励的输入周期、第二指令计数器激励的输入周期、...、第N指令计数器激励的输入周期依次增大,N为正整数;

确定所述寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于所述第一指令计数器激励的输入周期,得到第一确定结果;以及

根据所述第一确定结果,依次执行所述第一寄存器配置激励和所述第一指令计数器激励,得到针对所述第一寄存器配置激励和所述第一指令计数器激励的第二输出结果。

5. 根据权利要求4所述的验证方法,其特征在于,所述根据所述第一确定结果,依次执行所述第一寄存器配置激励和所述第一指令计数器激励,得到针对所述第一寄存器配置激励和所述第一指令计数器激励的第二输出结果,包括:

在所述第一确定结果指示了所述寄存器配置激励中存在未执行的第一寄存器配置激励的输入周期小于所述第一指令计数器激励的输入周期的情况下,依次执行所述第一寄存器配置激励和所述第一指令计数器激励,得到针对所述第一寄存器配置激励和所述第一指令计数器激励的第二输出结果;以及

在所述第一确定结果指示了所述寄存器配置激励中不存在未执行的第一寄存器配置激励的输入周期小于所述第一指令计数器激励的输入周期的情况下,依次执行所述第一指令计数器激励,得到针对所述第一指令计数器激励的第二输出结果。

6. 根据权利要求4所述的验证方法,其特征在于,该方法还包括:

确定所述寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于所述第二指令计数器激励的输入周期,得到第二确定结果;以及

根据所述第二确定结果,依次执行所述第二寄存器配置激励和所述第二指令计数器激励,得到针对所述第二寄存器配置激励和所述第二指令计数器激励的第二输出结果。

7. 根据权利要求4所述的验证方法,其特征在于,该方法还包括:

确定所述寄存器配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于所

述第N指令计数器激励的输入周期,得到第N确定结果;以及

根据所述第N确定结果,依次执行所述第N寄存器配置激励和所述第N指令计数器激励,得到针对所述第N寄存器配置激励和所述第N指令计数器激励的第二输出结果。

8. 一种追踪编码器的验证装置,其特征在于,包括:

获取单元,配置为获取追踪编码器的输入激励的第一输出结果;其中,所述第一输出结果为对所述输入激励进行被测追踪编码生成的输出结果;

激励预处理单元,配置为对所述输入激励进行预处理,得到预处理后的输入激励;

模拟追踪编码单元,配置为对所述预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及

结果分析单元,配置为根据所述第一输出结果和所述第二输出结果,得到针对所述输入激励的追踪编码结果。

9. 根据权利要求8所述的验证装置,其特征在于,所述输入激励包括指令计数器激励和寄存器配置激励;其中,所述激励预处理单元配置为:

根据所述指令计数器激励的输入时刻和所述寄存器配置激励的输入时刻,生成与所述指令计数器激励和所述寄存器配置激励分别对应的输入周期。

10. 根据权利要求9所述的验证装置,其特征在于,所述模拟追踪编码单元配置为:

获取预处理后的指令计数器激励中未执行的N个指令计数器激励;其中,第一指令计数器激励的输入周期、第二指令计数器激励的输入周期、...、第N指令计数器激励的输入周期依次增大,N为正整数;

确定所述寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于所述第一指令计数器激励的输入周期,得到第一确定结果;以及

根据所述第一确定结果,依次执行所述第一寄存器配置激励和所述第一指令计数器激励,得到针对所述第一寄存器配置激励和所述第一指令计数器激励的第二输出结果。

11. 根据权利要求10所述的验证装置,其特征在于,所述模拟追踪编码单元还配置为:

确定所述寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于所述第二指令计数器激励的输入周期,得到第二确定结果;以及

根据所述第二确定结果,依次执行所述第二寄存器配置激励和所述第二指令计数器激励,得到针对所述第二寄存器配置激励和所述第二指令计数器激励的第二输出结果。

12. 根据权利要求10所述的验证装置,其特征在于,所述模拟追踪编码单元还配置为:

确定所述寄存器配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于所述第N指令计数器激励的输入周期,得到第N确定结果;以及

根据所述第N确定结果,依次执行所述第N寄存器配置激励和所述第N指令计数器激励,得到针对所述第N寄存器配置激励和所述第N指令计数器激励的第二输出结果。

13. 一种电子设备,其特征在于,包括:

追踪编码被测单元,配置为对输入激励进行被测追踪编码,得到第一输出结果;

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1至7中任一项所述的验证方法。

14. 一种存储有计算机指令的非瞬时计算机可读存储介质,其中,所述计算机指令用于使所述计算机执行根据权利要求1至7中任一项所述的验证方法。

15. 一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现根据权利要求1至7中任一项所述的验证方法。

追踪编码器的验证方法、装置和电子设备

技术领域

[0001] 本公开涉及追踪编码器技术领域,更具体地,涉及一种追踪编码器的验证方法、装置和电子设备。

背景技术

[0002] 追踪编码器(Trace Encoder,TE)根据Trace协议(如,Nexus 5001 ForumTM Standard、Efficient Trace for RISC-V)将核心Core退休指令计数器(Program Counter,PC)和与Core退休PC相关的属性生成编码数据包。追踪解码器(Trace Decoder,TD)根据Trace协议解析编码数据包生成PC。

[0003] TE电路可以根据实际应用场景实时配置寄存器,以控制电路的功能。例如,在周期时间A内,通过配置寄存器更新指令地址,实现过滤条件和更新模式。但是,基于C语言或Python语言的TE模型(TE Model)与基于System Verilog语言的TE模型不同。基于C语言或Python语言的TE Model不能实时接收输入激励与模拟电路行为,将仿真过程中所有的输入激励直接加载到程序模型里,不能将各输入激励分时段依次加载到程序模型中。因此,TE Model在加载激励文件时无法识别各输入激励的先后顺序,无法依次模拟电路设计功能,不能得到模型期望结果。

发明内容

[0004] 本公开提出了一种追踪编码器的验证方法、装置和电子设备。

[0005] 根据本公开的第一方面,提出了一种追踪编码器的验证方法,包括:获取追踪编码器的输入激励的第一输出结果;其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果;对输入激励进行预处理,得到预处理后的输入激励;对预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及,根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。

[0006] 例如,输入激励包括指令计数器激励和寄存器配置激励;其中,对输入激励进行预处理,得到预处理后的输入激励,包括:根据指令计数器激励的输入时刻和寄存器配置激励的输入时刻,生成与指令计数器激励和寄存器配置激励分别对应的输入周期。

[0007] 例如,预处理后的输入激励中每个输入激励的输入周期彼此不同。

[0008] 例如,对预处理后的输入激励进行模拟追踪编码,得到第二输出结果,包括:获取预处理后的指令计数器激励中未执行的N个指令计数器激励;其中,第一指令计数器激励的输入周期、第二指令计数器激励的输入周期、...、第N指令计数器激励的输入周期依次增大,N为正整数;确定寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于第一指令计数器激励的输入周期,得到第一确定结果;以根据第一确定结果,依次执行第一寄存器配置激励和第一指令计数器激励,得到针对第一寄存器配置激励和第一指令计数器激励的第二输出结果。

[0009] 例如,根据第一确定结果,依次执行第一寄存器配置激励和第一指令计数器激励,

得到针对第一寄存器配置激励和第一指令计数器激励的第二输出结果,包括:在第一确定结果指示了寄存器配置激励中存在未执行的第一寄存器配置激励的输入周期小于第一指令计数器激励的输入周期的情况下,依次执行第一寄存器配置激励和第一指令计数器激励,得到针对第一寄存器配置激励和第一指令计数器激励的第二输出结果;在第一确定结果指示了寄存器配置激励中不存在未执行的第一寄存器配置激励的输入周期小于第一指令计数器激励的输入周期的情况下,依次执行第一指令计数器激励,得到针对第一指令计数器激励的第二输出结果。

[0010] 例如,该方法还包括:确定寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于第二指令计数器激励的输入周期,得到第二确定结果;以及,根据第二确定结果,依次执行第二寄存器配置激励和第二指令计数器激励,得到针对第二寄存器配置激励和第二指令计数器激励的第二输出结果。

[0011] 例如,该方法还包括:确定配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于第N指令计数器激励的输入周期,得到第N确定结果;以及,根据第N确定结果,依次执行第N寄存器配置激励和第N指令计数器激励,得到针对第N寄存器配置激励和第N指令计数器激励的第二输出结果。

[0012] 根据本公开实施例的第二方面,提供了一种追踪编码器的验证装置,包括:获取单元,配置为获取追踪编码器的输入激励的第一输出结果;其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果;激励预处理单元,配置为对输入激励进行预处理,得到预处理后的输入激励;模拟追踪编码单元,配置为对预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及,结果分析单元,配置为根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。

[0013] 例如,输入激励包括指令计数器激励和寄存器配置激励;其中,激励预处理单元配置为:根据指令计数器激励的输入时刻和寄存器配置激励的输入时刻,生成与指令计数器激励和寄存器配置激励分别对应的输入周期。

[0014] 例如,模拟追踪编码单元配置为:获取预处理后的指令计数器激励中未执行的N个指令计数器激励;其中,第一指令计数器激励的输入周期、第二指令计数器激励的输入周期、...、第N指令计数器激励的输入周期依次增大,N为正整数;确定寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于第一指令计数器激励的输入周期,得到第一确定结果;以及,根据第一确定结果,依次执行第一寄存器配置激励和第一指令计数器激励,得到针对第一寄存器配置激励和第一指令计数器激励的第二输出结果。

[0015] 例如,模拟追踪编码单元还配置为:确定寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于第二指令计数器激励的输入周期,得到第二确定结果;以及,根据第二确定结果,依次执行第二寄存器配置激励和第二指令计数器激励,得到针对第二寄存器配置激励和第二指令计数器激励的第二输出结果。

[0016] 例如,模拟追踪编码单元还配置为:确定寄存器配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于第N指令计数器激励的输入周期,得到第N确定结果;以及,根据第N确定结果,依次执行第N寄存器配置激励和第N指令计数器激励,得到针对第N寄存器配置激励和第N指令计数器激励的第二输出结果。

[0017] 根据本公开实施例的第三方面,提供了一种电子设备,包括:追踪编码被测单元,

配置为对输入激励进行被测追踪编码,得到第一输出结果;至少一个处理器;以及与至少一个处理器通信连接的存储器;其中,存储器存储有可被至少一个处理器执行的指令,指令被至少一个处理器执行,以使至少一个处理器能够执行根据本公开提供的方法。

[0018] 根据本公开实施例的第四方面,提供了一种存储有计算机指令的非瞬时计算机可读存储介质,该计算机指令用于使计算机执行根据本公开提供的方法。

[0019] 根据本公开实施例的第五方面,提供了一种计算机程序产品,包括计算机程序,该计算机程序在被处理器执行时实现根据本公开提供的方法。

[0020] 根据公开实施例的技术方案,提供了一种追踪编码器的验证方法。该方法将各输入激励添加时间周期属性,使得TE Model对各输入激励进行模拟追踪编码时,能够根据输入周期属性识别各输入激励的先后顺序,并依次执行各输入激励。该方法能够降低电路功能的耦合性,提高TE Model的可重用性,无需增加TE Model复杂度。此外,该方法缩短了电路功能的验证周期,减少验证成本,有利于提前发现电路异常。

附图说明

[0021] 通过下面结合附图说明本公开实施例,将使本公开实施例的上述及其它目的、特征和优点更加清楚。应注意,贯穿附图,相同的元素由相同或相近的附图标记来表示。图中:

[0022] 图1示出了根据本公开一实施例的追踪编码器的验证方法的应用场景示意图;

[0023] 图2示出了根据本公开一实施例的追踪编码器的验证方法的流程图;

[0024] 图3A和图3B分别示出了根据本公开一实施例的预处理前和预处理后的指令计数器激励的示意图;

[0025] 图4A和图4B分别示出了根据本公开一实施例的预处理前和预处理后的寄存器配置激励的示意图;

[0026] 图5示出了根据本公开一实施例的对预处理后的输入激励进行模拟追踪编码的流程图;

[0027] 图6示出了根据本公开一实施例的追踪编码器的验证装置的结构框图;

[0028] 图7示出了根据本公开一实施例的可以应用追踪编码器的验证方法的电子设备的框图。

具体实施方式

[0029] 为使本公开实施例的目的、技术方案和优点更加清楚,下面将结合本公开实施例中的附图,对本公开实施例中的技术方案进行清楚、完整的描述。显然,所描述的实施例是本公开的一部分实施例,而不是全部。基于所描述的本公开实施例,本领域普通技术人员在无需创造性劳动的前提下获得的所有其他实施例都属于本公开保护的范围。在以下描述中,一些具体实施例仅用于描述目的,而不应理解为对本公开有任何限制,而只是本公开实施例的示例。在可能导致对本公开的理解造成混淆时,将省略常规结构或构造。应注意,图中各部件的形状和尺寸不反映真实大小和比例,而仅示意本公开实施例的内容。

[0030] 除非另外定义,本公开实施例使用的技术术语或科学术语应当是本领域技术人员所理解的通常意义。本公开实施例中使用的“第一”、“第二”、“第三”以及类似词语并不表示任何顺序、数量或重要性,而只是用于区分不同的组成部分。

[0031] 本公开提供了一种追踪编码器的验证方法、装置、电子设备、存储介质以及程序产品。

[0032] 根据本公开的实施例,提供了一种追踪编码器的验证方法,包括:获取追踪编码器的输入激励的第一输出结果;其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果;对输入激励进行预处理,得到预处理后的输入激励;对预处理后的输入激励进行模拟追踪编码,得到第二输出结果;以及,根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。

[0033] 图1示意性示出了根据本公开实施例的可以应用于追踪编码器的验证方法的示例性系统架构100。需要注意的是,图1所示仅为可以应用本公开实施例的系统架构的示例,以帮助本领域技术人员理解本公开的技术内容,但并不意味着本公开实施例不可以用于其他设备、系统、环境或场景。

[0034] 如图1所示,根据该实施例的系统架构100包括追踪编码被测模块(Trace Encoder Design Under Test, TE DUT) 110、激励预处理模块120、TE Model 130和结果分析模块140。

[0035] TE DUT 110为TE的电路模块,例如电源电路、光电转换电路、信号处理电路等。系统验证工程师将输入激励输入至TE DUT 110中,使TE DUT 110对该输入激励进行被测追踪编码,得到实际输出结果。

[0036] 例如,输入激励可以为指令计数器激励(PC激励)和寄存器配置激励。PC激励指示存放当前欲执行指令的地址。寄存器配置激励用于执行寄存器的某一功能,以完成电路功能检测。

[0037] 激励预处理模块120,用于对输入激励进行预处理,得到预处理后的输入激励。

[0038] 例如,对输入激励进行预处理的过程可以为将程序输入文件中的各输入激励添加输入周期属性(Cycle Time)。例如,根据各输入激励的输入时刻,生成与各输入激励对应的输入周期。

[0039] 例如,TE DUT 110的寄存器配置场景不支持在PC激励输入的相同周期内,进行寄存器配置激励的输入。因此,每个PC激励的Cycle Time和每个寄存器配置激励的Cycle Time彼此不同。

[0040] TE Model 130,用于对预处理后的输入激励进行模拟追踪编码,得到第二输出结果。

[0041] 例如,第二输出结果可以为期望输出结果,用以验证寄存器的功能实现是否正常。

[0042] 例如,TE Model 130可以为C语言或Python语言实现的TE Model。

[0043] 结果分析模块140,用于对实际输出结果和期望输出结果进行对比分析,输出针对输入激励的追踪编码结果。

[0044] 例如,结果分析模块140可以通过脚本实现,用于对输出结果进行自动对比分析。

[0045] 例如,在实际输出结果和期望输出结果相同的情况下,该追踪编码结果指示TE的电路功能正常。在实际输出结果和期望输出结果不同的情况下,该追踪编码结果指示TE的电路功能存在异常。

[0046] 应当理解,图1中的TE DUT、激励预处理模块、TE Model和结果分析模块的连接关系及数目仅仅是示意性的。根据实现需要,可以具有任意数目的TE DUT、激励预处理模块、TE Model和结果分析模块。

[0047] 图2示出了根据本公开一实施例的追踪编码器的验证方法的流程图。

[0048] 如图2所示,该追踪编码器的验证方法200,包括操作S210~S240。

[0049] 在操作S210,获取追踪编码器的输入激励的第一输出结果。其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果。

[0050] 在本公开实施例中,该验证方法200可以由TE DUT 110执行。系统验证工程师可以向TE DUT 110发送用于执行电路功能的输入激励。TE DUT 110接收执行电路功能的输入激励,并执行该输入激励,输出第一输出结果。

[0051] 例如,输入激励可以为PC激励和寄存器配置激励。PC激励指示存放当前欲执行指令的地址。寄存器配置激励用于执行寄存器的某一功能,以完成电路功能检测。

[0052] 例如,输入激励可以为一个或多个输入激励。

[0053] 例如,第一输出结果可以为对电路功能进行被测验证的实际输出结果。第一输出结果可以指示电路功能正常或电路功能异常。

[0054] 在操作S220,对输入激励进行预处理,得到预处理后的输入激励。

[0055] 在本公开实施例中,对输入激励进行预处理的过程可以为在存储有各输入激励的程序输入文件中,为每个输入激励添加Cycle Time属性。

[0056] 例如,根据每个输入激励的输入时刻,生成与该输入激励对应的Cycle Time。

[0057] 例如,输入激励中第一输入激励的输入时刻为a1时b1分c1秒,第一输入激励对应的Cycle Time为A。A表示第一输入激励相距前一时刻输入激励0的时钟周期差。类似地,输入激励中第二输入激励的输入时刻为a1时b1分c2秒,第二输入激励对应的Cycle Time为B。B表示第二输入激励相距输入激励0的时钟周期差。基于此,每个输入激励的Cycle Time彼此不同。

[0058] 需说明的是,上述实施例中的输入时刻和Cycle Time仅为示例性的说明,以便于本领域技术人员理解本公开的方案,并不构成本公开实施例的限定。

[0059] 在操作S230,对预处理后的输入激励进行模拟追踪编码,得到第二输出结果。

[0060] 例如,预处理后的PC激励包括第一PC激励、第二PC激励、...、第K PC激励。第一PC激励、第二PC激励、...、第K PC激励的Cycle Time依次增大。K为大于等于2的整数。

[0061] 在本公开实施例中,对预处理后的输入激励进行模拟追踪编码,可以为获取预处理后的输入激励中未执行的第一PC激励,该第一PC激励在预处理后的PC激励中Cycle Time最小。通过确定预处理后的寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期。根据确定结果,依次执行第一寄存器配置激励和第一PC激励,得到针对第一寄存器配置激励和第一PC激励的第二输出结果。

[0062] 例如,在依次执行第一寄存器配置激励和第一PC激励之后,继续确定预处理后的寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期。根据确定结果,依次执行第二寄存器配置激励和第二PC激励,得到针对第二寄存器配置激励和第二PC激励的第二输出结果。

[0063] 例如,以此类推,在依次执行第K-1寄存器配置激励和第K-1 PC激励之后,继续确定预处理后的寄存器配置激励中是否存在未执行的第K寄存器配置激励的输入周期小于第K PC激励的输入周期。根据确定结果,依次执行第K寄存器配置激励和第K PC激励,得到针对第K寄存器配置激励和第K PC激励的第二输出结果,直至完成对程序输入文件中所有输

入激励的模拟追踪编码。

[0064] 例如,在每次执行完部分输入激励后,可以将这些已执行的输入激励的状态更新为“已执行”,以便于识别程序输入文件中未执行的输入激励和已执行的输入激励。

[0065] 例如,第二输出结果为对各输入激励的期望输出结果,以用于与实际输出结果进行对比分析,进而发现TE电路功能存在的异常。

[0066] 在操作S240,根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。

[0067] 在本公开实施例中,将第一输出结果和第二输出结果进行对比分析,具体为将第一输出结果中第k寄存器配置激励的执行结果和第二输出结果中第k寄存器配置激励的执行结果进行对比分析,得到针对第k寄存器配置激励的追踪编码结果。 $1 \leq k \leq K$, k为整数。

[0068] 例如,在第一输出结果中第k寄存器配置激励的执行结果和第二输出结果中第k寄存器配置激励的执行结果相同的情况下,该追踪编码结果指示第k寄存器配置对应的电路功能正常。在第一输出结果中第k寄存器配置激励的执行结果和第二输出结果中第k寄存器配置激励的执行结果不同的情况下,该追踪编码结果指示第k寄存器配置对应的电路功能存在异常。

[0069] 根据本公开的实施例,通过逐一比较第一输出结果中每个寄存器配置激励的执行结果和第二输出结果中每个寄存器配置激励的执行结果,可以得到TE电路模块的功能验证结果,以便于发现TE电路模块存在的异常。

[0070] 本公开的实施例提供的方法,将各输入激励添加时间周期属性,使得TE Model对各输入激励进行模拟追踪编码时,能够根据输入周期属性识别各输入激励的先后顺序,并依次执行各输入激励。该方法能够降低电路功能的耦合性,提高TE Model的可重用性,无需增加TE Model复杂度。此外,该方法缩短了电路功能的验证周期,减少验证成本,有利于提前发现电路异常。

[0071] 图3A和图3B分别示出了根据本公开一实施例的预处理前和预处理后的指令计数器激励的示意图。图4A和图4B分别示出了根据本公开一实施例的预处理前和预处理后的寄存器配置激励的示意图。

[0072] 根据本公开的实施例,对输入激励进行预处理,得到预处理后的输入激励,包括根据PC激励的输入时刻和寄存器配置激励的输入时刻,生成与PC激励和寄存器配置激励分别对应的输入周期。

[0073] 如图3A所示,程序输入文件300A存储有未经预处理的多个PC激励,分别为8000_0000、8000_0004、8000_0008、...、8000_0080。xx表示每个PC激励对应指向的数据,本公开的实施例对此不做限定。

[0074] 如图3B所示,程序输入文件300B存储有预处理后的多个PC激励,分别为8000_0000(cycle_time:12)、8000_0004(cycle_time:15)、8000_0008(cycle_time:17)、...、8000_0080(cycle_time:98)。在图3B的示例中,每个预处理后的PC激励的Cycle Time彼此不同。

[0075] 如图4A所示,程序输入文件400A存储有未经预处理的多个寄存器配置激励,分别为Addr:'h0010、Addr:'h0014、...、Addr:'h0014、...、Addr:'h0018、Addr:'h0010。xx表示每个寄存器配置激励执行时的配置数据,本公开的实施例对此不做限定。

[0076] 例如,每个寄存器配置激励对应的配置数据可以为写入数据类型或读取数据类

型,可以通过每个寄存器配置激励对应的配置数据data的属性RW来识别数据类型。例如,RW=0,代表读取数据。RW=1,代表写入数据。

[0077] 如图4B所示,程序输入文件400B存储有预处理后的多个寄存器配置激励,分别为Addr:'h0010(cycle_time:9)、Addr:'h0014(cycle_time:11)、...、Addr:'h0014(cycle_time:61)、...、Addr:'h0018(cycle_time:70)、Addr:'h0010(cycle_time:96)。在图4B的示例中,每个预处理后的寄存器配置激励的Cycle Time彼此不同。

[0078] 在本公开实施例中,TE DUT的寄存器配置场景不支持在PC激励输入的相同周期内,进行寄存器配置激励的输入。因此,每个PC激励和每个寄存器配置激励的Cycle Time彼此不同。

[0079] 例如,输入激励预处理的过程可以通过脚本实现,本公开的实施例对此不做限定。

[0080] 需说明的是,上述实施例中每个输入激励的属性和Cycle Time仅为示例性的说明,并不构成本公开实施例的限定。

[0081] 图5示出了根据本公开一实施例的对预处理后的输入激励进行模拟追踪编码的流程图。

[0082] 如图5所示,对预处理后的输入激励进行模拟追踪编码过程,包括操作S510~S550。

[0083] 在操作S510,获取预处理后的PC激励中未执行的N个PC激励。N个PC激励中第一PC激励的输入周期、第二PC激励的输入周期、...、第N PC激励的输入周期依次增大。N为正整数。

[0084] 例如,每个预处理后的PC激励具有Cycle Time属性。Cycle Time属性用于识别每个预处理后的PC激励的先后顺序,以使能够对多个PC激励依次执行处理。

[0085] 在操作S520,确定寄存器配置激励中是否存在未执行的第n寄存器配置激励的输入周期小于第n PC激励的输入周期,得到第n确定结果。 $1 \leq n \leq N$,n为整数。

[0086] 例如,当n=1时,确定寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期,得到第一确定结果。该第一确定结果指示存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期或不存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期。

[0087] 例如,沿用如图3B所示的预处理后的多个PC激励的示例,以及沿用如图4B所示的预处理后的多个寄存器配置激励的示例。其中,PC激励8000_0000的输入周期最小,则将PC激励8000_0000为第一输入激励。如图4B所示的预处理后的多个寄存器配置激励中Addr:'h0010的输入周期和Addr:'h0014的输入周期小于PC激励8000_0000的输入周期。因此,第一确定结果指示存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期,第一寄存器配置激励包括Addr:'h0010和Addr:'h0014。

[0088] 例如,当n=2时,确定寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期,得到第二确定结果。该第二确定结果指示存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期或不存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期。

[0089] 例如,当n=N时,确定寄存器配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于第N PC激励的输入周期,得到第N确定结果。该第N确定结果指示存在未执行

的第N寄存器配置激励的输入周期小于第N PC激励的输入周期或不存在未执行的第N寄存器配置激励的输入周期小于第N PC激励的输入周期。

[0090] 在操作S530,在第n确定结果指示了寄存器配置激励中存在未执行的第n寄存器配置激励的输入周期小于第n PC激励的输入周期的情况下,依次执行第n寄存器配置激励和第n PC激励。

[0091] 在本公开实施例中,在第n确定结果指示了寄存器配置激励中存在未执行的第n寄存器配置激励的输入周期小于第n PC激励的输入周期的情况下,按照输入周期大小依次执行第n寄存器配置激励中的各寄存器配置激励之后,再执行第n PC激励。

[0092] 例如,在第一确定结果指示了寄存器配置激励中存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期的情况下,按照输入周期大小依次执行第一寄存器配置激励中的各寄存器配置激励之后,再执行第一PC激励。

[0093] 例如,沿用上述实施例,按照输入周期大小依次执行第一寄存器配置激励中Addr:'h0010和Addr:'h0014,再执行PC激励80000000。

[0094] 例如,在Addr:'h0010、Addr:'h0014和80000000被执行之后,可将Addr:'h0010、Addr:'h0014和80000000的状态分别更新为“已执行”,以便于识别程序输入文件中未执行的输入激励和已执行的输入激励Addr:'h0010、Addr:'h0014和80000000。

[0095] 例如,在第二确定结果指示了寄存器配置激励中存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期的情况下,按照输入周期大小依次执行第二寄存器配置激励中的各寄存器配置激励之后,再执行第二PC激励。

[0096] 例如,在第N确定结果指示了寄存器配置激励中存在未执行的第N寄存器配置激励的输入周期小于第N PC激励的输入周期的情况下,按照输入周期大小依次执行第N寄存器配置激励中的各寄存器配置激励之后,再执行第N PC激励。

[0097] 在操作S540,在第n确定结果指示了寄存器配置激励中不存在未执行的第n寄存器配置激励的输入周期小于第n PC激励的输入周期的情况下,执行第n PC激励。

[0098] 例如,在第一确定结果指示了寄存器配置激励中不存在未执行的第一寄存器配置激励的输入周期小于第一PC激励的输入周期的情况下,执行第一PC激励。

[0099] 例如,在第二确定结果指示了寄存器配置激励中不存在未执行的第二寄存器配置激励的输入周期小于第二PC激励的输入周期的情况下,执行第二PC激励。

[0100] 例如,在第N确定结果指示了寄存器配置激励中不存在未执行的第N寄存器配置激励的输入周期小于第N PC激励的输入周期的情况下,执行第N PC激励。

[0101] 在操作S550,根据依次执行第n寄存器配置激励和第n PC激励的执行结果或执行第n PC激励的执行结果,生成针对第n寄存器配置激励和第n PC激励的第二输出结果或针对第n PC激励的第二输出结果。

[0102] 例如,在执行完第n寄存器配置激励和第n PC激励后,将生成的编码数据包写入前一时刻生成的第二输出结果中,生成针对第n寄存器配置激励和第n PC激励的第二输出结果。或,在执行完第n PC激励后,将生成的编码数据包写入前一时刻生成的第二输出结果中,生成针对第n PC激励的第二输出结果。

[0103] 例如,每个输入激励的执行过程可以包括执行该输入激励和与该输入激励对应的属性。输入激励属性可以为指令参数、指令类型、指令长度等。

[0104] 在本公开实施例中,在依次执行完第n寄存器配置激励和第n PC激励的执行结果之后,或执行完第n PC激励之后,获取程序输入文件中比第n PC激励的输入周期时间次小的第n+1PC激励。确定寄存器配置激励中是否存在未执行的第n+1寄存器配置激励的输入周期小于第n+1PC激励的输入周期,得到第n+1确定结果。根据第n+1确定结果,依次执行第n+1寄存器配置激励和第n+1PC激励,并生成相应的第二输出结果。以此类推,完成对所有寄存器配置激励和所有PC激励的执行,生成第二输出结果。

[0105] 本公开的实施例提供的方法,该方法提供一种基于C语言或Python语言的TE模型应用场景下,对输入激励进行输入周期属性配置,使得基于C语言或Python语言的TE Model无法识别初始输入激励先后顺序的情况下,可以根据输入周期属性识别每个输入周期的先后顺序,以按照输入周期大小依次执行输入激励,完成对TE电路功能模块的验证。该方法实现的TE Model复杂度降低,可重用性高,缩短了验证周期,有利于提前发现TE电路异常。

[0106] 图6示出了根据本公开一实施例的追踪编码器的验证装置的结构框图。

[0107] 如图6所示,该追踪编码器的验证600可以实现方法200,包括:获取单元610、激励预处理单元620、模拟追踪编码单元630和结果分析单元640。

[0108] 获取单元610,配置为获取追踪编码器的输入激励的第一输出结果。其中,第一输出结果为对输入激励进行被测追踪编码生成的输出结果。

[0109] 例如,第一获取单元610用于执行上述实施例中的操作S210,不再赘述。

[0110] 激励预处理单元620,配置为对输入激励进行预处理,得到预处理后的输入激励。

[0111] 例如,激励预处理单元620用于执行上述实施例中的操作S220,不再赘述。

[0112] 模拟追踪编码单元630,配置为对预处理后的输入激励进行模拟追踪编码,得到第二输出结果。

[0113] 例如,模拟追踪编码单元630用于执行上述实施例中的操作S230,不再赘述。

[0114] 结果分析单元640,配置为根据第一输出结果和第二输出结果,得到针对输入激励的追踪编码结果。

[0115] 例如,结果分析单元640用于执行上述实施例中的操作S240,不再赘述。

[0116] 根据本公开的实施例,输入激励包括指令计数器激励和寄存器配置激励。激励预处理单元620配置为根据指令计数器激励的输入时刻和寄存器配置激励的输入时刻,生成与指令计数器激励和寄存器配置激励分别对应的输入周期。

[0117] 根据本公开的实施例,模拟追踪编码单元630配置为获取预处理后的指令计数器激励中未执行的N个指令计数器激励;其中,第一指令计数器激励的输入周期、第二指令计数器激励的输入周期、...、第N指令计数器激励的输入周期依次增大,N为正整数;确定寄存器配置激励中是否存在未执行的第一寄存器配置激励的输入周期小于第一指令计数器激励的输入周期,得到第一确定结果;以及,根据第一确定结果,依次执行第一寄存器配置激励和第一指令计数器激励,得到针对第一寄存器配置激励和第一指令计数器激励的第二输出结果。

[0118] 根据本公开的实施例,模拟追踪编码单元630还配置为确定寄存器配置激励中是否存在未执行的第二寄存器配置激励的输入周期小于第二指令计数器激励的输入周期,得到第二确定结果;以及,根据第二确定结果,依次执行第二寄存器配置激励和第二指令计数器激励,得到针对第二寄存器配置激励和第二指令计数器激励的第二输出结果。

[0119] 根据本公开的实施例,模拟追踪编码单元630还配置为确定寄存器配置激励中是否存在未执行的第N寄存器配置激励的输入周期小于第N指令计数器激励的输入周期,得到第N确定结果;以及,根据第N确定结果,依次执行第N寄存器配置激励和第N指令计数器激励,得到针对第N寄存器配置激励和第N指令计数器激励的第二输出结果。

[0120] 需说明的是,根据本公开的实施例的模块、子模块、单元、子单元中的任意多个、或其中任意多个的至少部分功能可以在一个模块中实现。根据本公开实施例的模块、子模块、单元、子单元中的任意一个或多个可以被拆分成多个模块来实现。根据本公开实施例的模块、子模块、单元、子单元中的任意一个或多个可以至少被部分地实现为硬件电路,例如现场可编程门阵列(FPGA)、可编程逻辑阵列(PLA)、片上系统、基板上的系统、封装上的系统、专用集成电路(ASIC),或可以通过对电路进行集成或封装的任何其他的合理方式的硬件或固件来实现,或以软件、硬件以及固件三种实现方式中任意一种或以其中任意几种的适当组合来实现。或者,根据本公开实施例的模块、子模块、单元、子单元中的一个或多个可以至少被部分地实现为计算机程序模块,当该计算机程序模块被运行时,可以执行相应的功能。

[0121] 例如,获取单元610、激励预处理单元620、模拟追踪编码单元630和结果分析单元640中的任意多个可以合并在一个模块中实现,或者其中的任意一个模块可以被拆分成多个模块。或者,这些模块中的一个或多个模块的至少部分功能可以与其他模块的至少部分功能相结合,并在一个模块中实现。根据本公开的实施例,获取单元610、激励预处理单元620、模拟追踪编码单元630和结果分析单元640中的至少一个可以至少被部分地实现为硬件电路,例如现场可编程门阵列(FPGA)、可编程逻辑阵列(PLA)、片上系统、基板上的系统、封装上的系统、专用集成电路(ASIC),或可以通过对电路进行集成或封装的任何其他的合理方式等硬件或固件来实现,或以软件、硬件以及固件三种实现方式中任意一种或以其中任意几种的适当组合来实现。或者,获取单元610、激励预处理单元620、模拟追踪编码单元630和结果分析单元640中的至少一个可以至少被部分地实现为计算机程序模块,当该计算机程序模块被运行时,可以执行相应的功能。

[0122] 图7示出了根据本公开一实施例的可以应用追踪编码器的验证方法的电子设备的框图。图7示出的电子设备仅仅是一个示例,不应对本公开实施例的功能和使用范围带来任何限制。

[0123] 如图7所示,本实施例中所描述的电子设备700,包括:追踪编码被测单元701、至少一个处理器702、存储器703和总线704。追踪编码被测单元701、至少一个处理器702以及存储器703通过总线704彼此相连。

[0124] 追踪编码被测单元701配置为对输入激励进行被测追踪编码,得到第一输出结果。例如,追踪编码被测单元701可以为TE的电路模块,例如电源电路、光电转换电路、信号处理电路等。

[0125] 至少一个处理器702可以包括用于执行根据本公开实施例的方法流程的不同动作的单一处理单元或者是多个处理单元。

[0126] 存储器703与至少一个处理器702通信连接。在存储器703中,存储有电子设备700操作所需的各种程序和数据。

[0127] 例如,存储器703存储有可被至少一个处理器702执行的指令,该指令被至少一个处理器702执行,以使执行本公开实施例提供的验证方法。处理器702通过执行存储器703中

的程序来执行根据本公开实施例的方法流程的各种操作。需要注意,所述程序也可以存储在除存储器703以外的一个或多个存储器中。处理器702也可以通过执行存储在一个或多个存储器703中的程序来执行根据本公开实施例的方法流程的各种操作。

[0128] 根据本公开的实施例,根据本公开实施例的方法流程可以被实现为计算机软件程序。例如,本公开的实施例包括一种计算机程序产品,其包括承载在计算机可读存储介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分从网络上被下载和安装,和/或从可拆卸介质被安装。在该计算机程序被处理器701执行时,执行本公开实施例的系统中限定的上述功能。根据本公开的实施例,上文描述的系统、设备、装置、模块、单元等可以通过计算机程序模块来实现。

[0129] 本发明实施例还提供了一种计算机可读存储介质,该计算机可读存储介质可以是上述实施例中描述的设备/装置/系统中所包含的;也可以是单独存在,而未装配入该设备/装置/系统中。上述计算机可读存储介质承载有一个或者多个程序,当上述一个或者多个程序被执行时,实现根据本公开实施例的方法。

[0130] 根据本公开的实施例,计算机可读存储介质可以是非易失性的计算机可读存储介质,例如可以包括但不限于:便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开的实施例中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。例如,根据本公开的实施例,计算机可读存储介质可以包括上文描述的ROM和/或RAM和/或ROM和RAM以外的一个或多个存储器。

[0131] 需要说明的是,在本发明各个实施例中的各功能模块可以集成在一个处理模块中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来。

[0132] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0133] 本领域技术人员可以理解,本公开的各个实施例和/或权利要求中记载的特征可以进行多种组合和/或结合,即使这样的组合或结合没有明确记载于本公开中。特别地,在不脱离本公开精神和教导的情况下,本公开的各个实施例和/或权利要求中记载的特征可

以进行多种组合和/或结合。所有这些组合和/或结合均落入本公开的范围。

[0134] 尽管已经参照本公开的特定示例性实施例示出并描述了本公开,但是本领域技术人员应该理解,在不背离所附权利要求及其等同物限定的本公开的精神和范围的情况下,可以对本公开进行形式和细节上的多种改变。因此,本公开的范围不应该限于上述实施例,而是应该不仅由所附权利要求来进行确定,还由所附权利要求的等同物来进行限定。

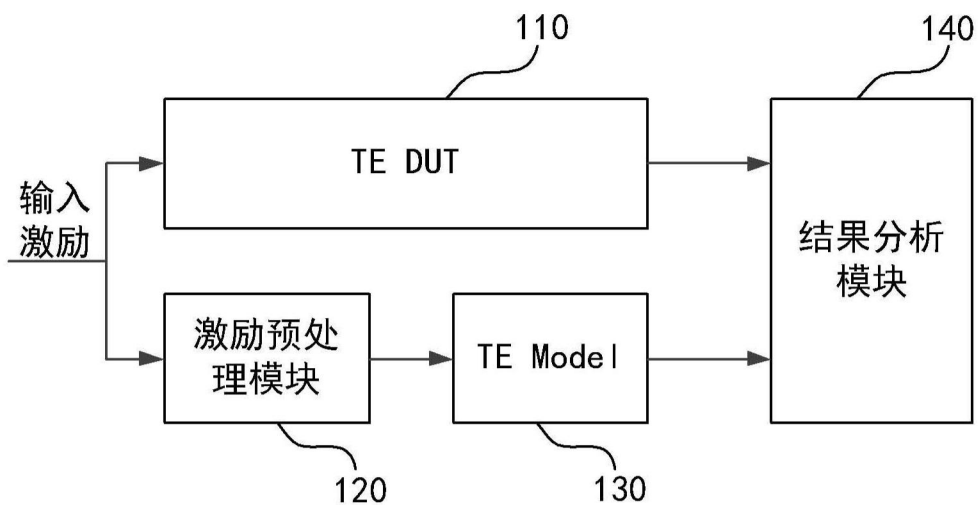
100

图1

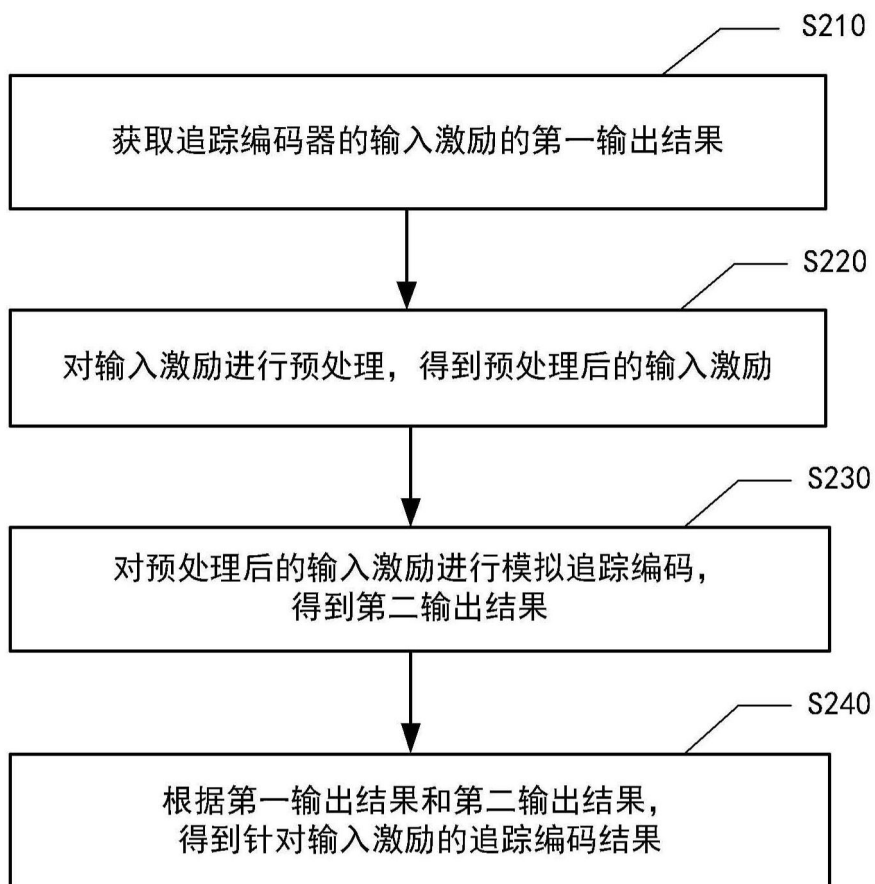
200

图2

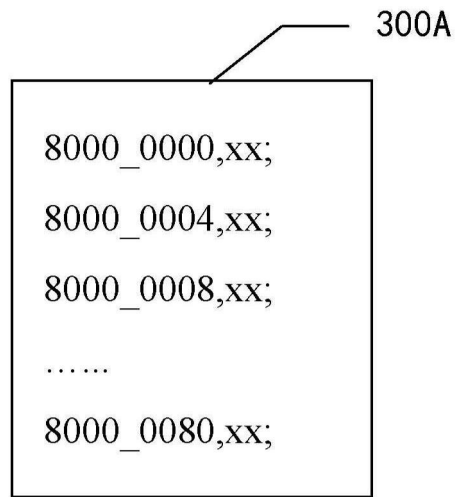


图3A

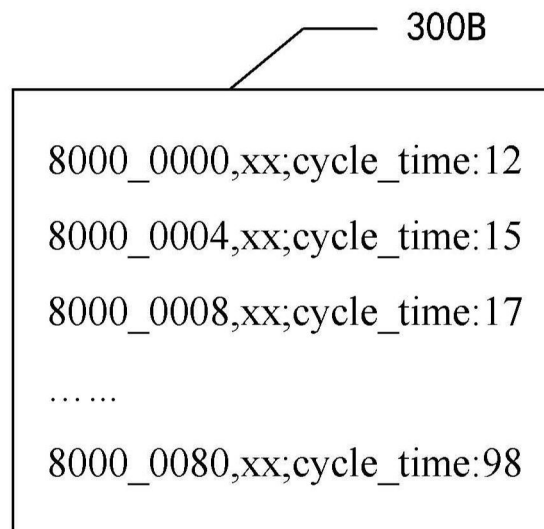


图3B

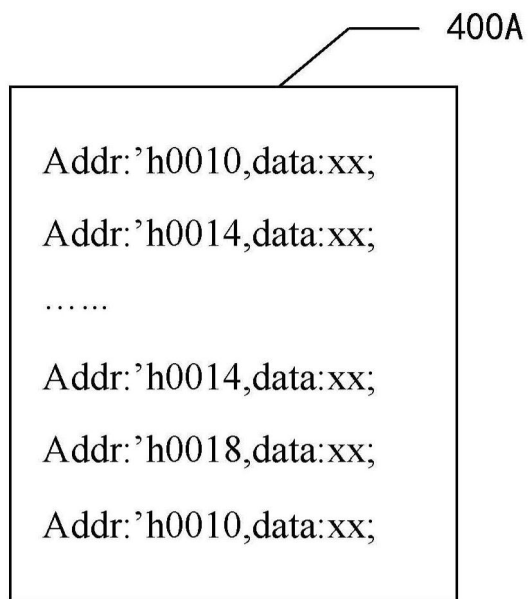


图4A

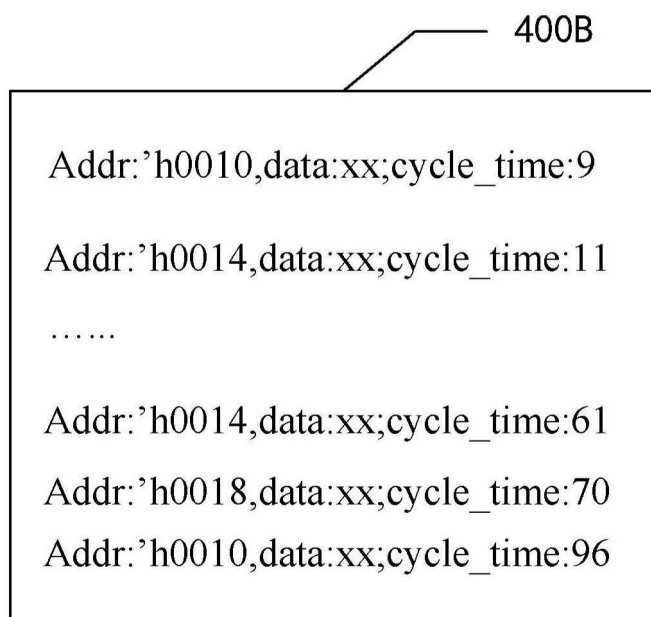


图4B

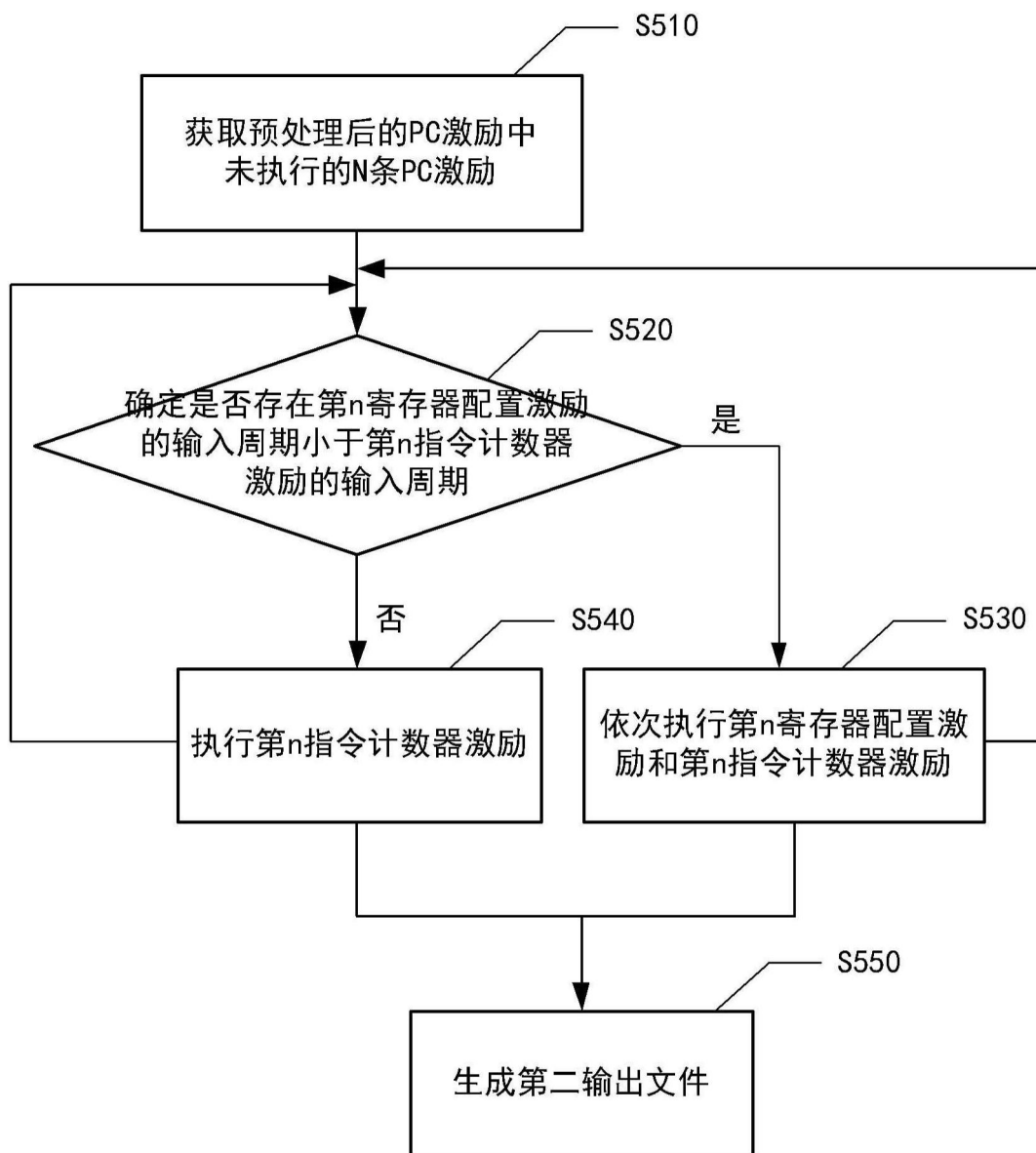


图5

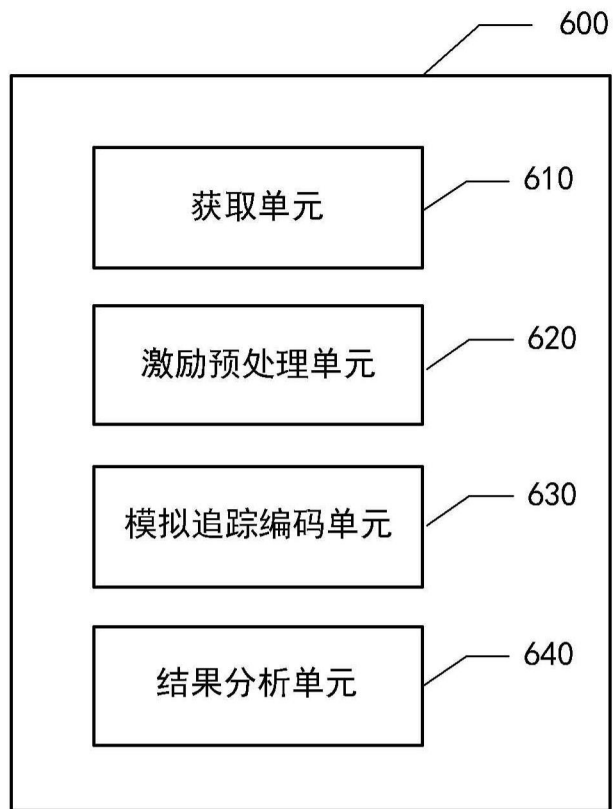


图6

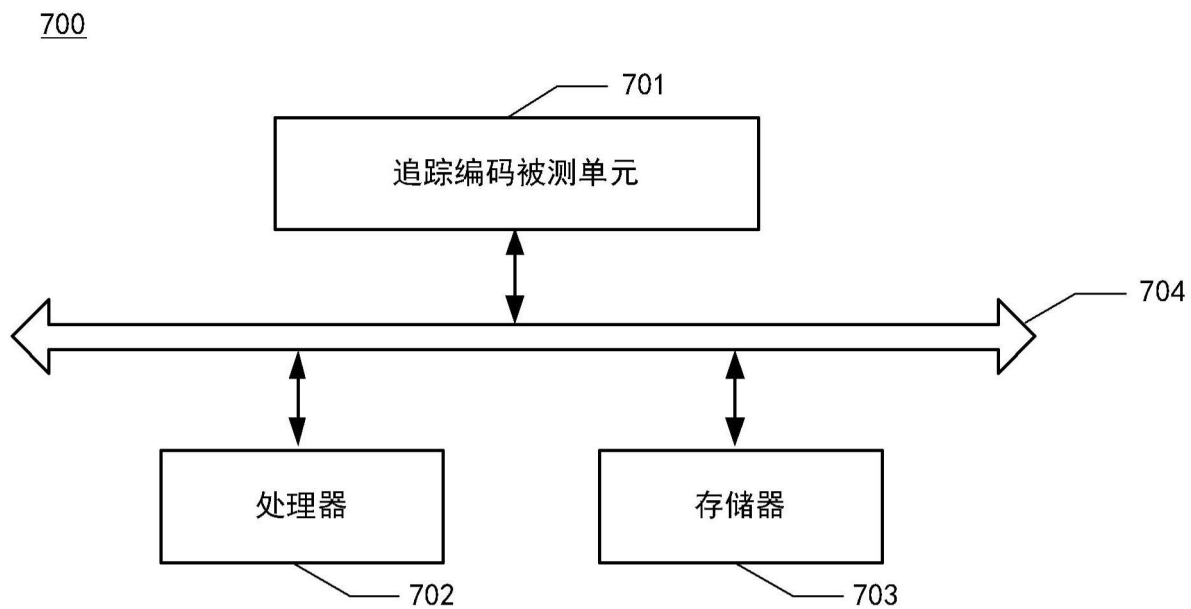


图7