



(12) 发明专利申请

(10) 申请公布号 CN 116501415 A

(43) 申请公布日 2023. 07. 28

(21) 申请号 202310787360.X

(22) 申请日 2023.06.30

(71) 申请人 英诺达(成都)电子科技有限公司
地址 610000 四川省成都市中国(四川)自由贸易试验区成都高新区和乐二街171号B6栋2单元8层803、804、818、819、820室

(72) 发明人 肖玉川

(74) 专利代理机构 北京铭硕知识产权代理有限公司 11286
专利代理师 王艳茹 苏银虹

(51) Int. Cl.
G06F 9/448 (2018.01)

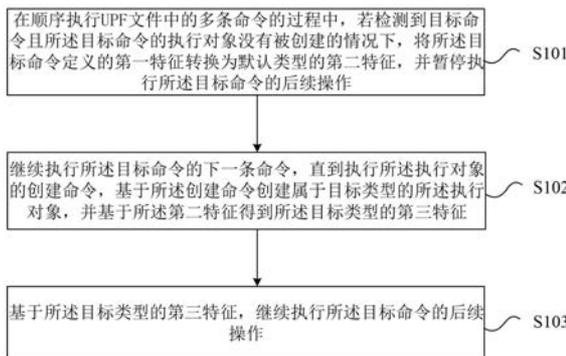
权利要求书3页 说明书16页 附图5页

(54) 发明名称

命令执行方法及装置、电子设备、计算机可读存储介质

(57) 摘要

本公开关于一种命令执行方法及装置、电子设备、计算机可读存储介质。方法包括：在顺序执行UPF文件中的多条命令的过程中，若检测到目标命令且目标命令的执行对象没有被创建的情况下，将目标命令定义的第一特征转换为默认类型的第二特征，并暂停执行目标命令的后续操作，其中，目标命令用于根据第一特征调整执行对象的特征；继续执行目标命令的下一条命令，直到执行到执行对象的创建命令，基于创建命令创建属于目标类型的执行对象，并基于第二特征得到目标类型的第三特征；基于目标类型的第三特征，继续执行目标命令的后续操作。



1. 一种命令执行方法,其特征在于,包括:

在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且所述目标命令的执行对象没有被创建的情况下,将所述目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行所述目标命令的后续操作,其中,所述目标命令用于根据所述第一特征调整所述执行对象的特征;

继续执行所述目标命令的下一条命令,直到执行所述执行对象的创建命令,基于所述创建命令创建属于目标类型的所述执行对象,并基于所述第二特征得到所述目标类型的第三特征;

基于所述目标类型的第三特征,继续执行所述目标命令的后续操作。

2. 如权利要求1所述的命令执行方法,其特征在于,通过如下方式确定所述目标命令的执行对象没有被创建:

确定所述目标命令的所述执行对象的类型信息;

将所述类型信息指示的类型依次与执行所述UPF文件的软件所支持的对象类型进行比对;

在比对结果均指示类型不一致的情况下,确定所述目标命令的所述执行对象没有被创建。

3. 如权利要求2所述的命令执行方法,其特征在于,在确定所述目标命令的执行对象没有被创建之后,还包括:

将最后与所述类型信息指示的类型比对的对象类型作为默认类型;

或者,指定执行所述UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。

4. 如权利要求1所述的命令执行方法,其特征在于,还包括:

若检测到所述目标命令且所述目标命令的执行对象已经被创建的情况下,基于所述目标命令定义的第一特征得到所述目标类型的第三特征;

基于所述目标类型的第三特征,继续执行所述目标命令的后续操作。

5. 如权利要求4所述的命令执行方法,其特征在于,通过如下方式确定所述目标命令的执行对象已经被创建:

确定所述目标命令的所述执行对象的类型信息;

将所述类型信息指示的类型依次与执行所述UPF文件的软件所支持的对象类型进行比对;

在所述类型信息指示的类型与执行所述UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定所述目标命令的执行对象已经被创建,并将所述任一个对象类型作为所述目标类型。

6. 如权利要求1所述的命令执行方法,其特征在于,所述基于所述第二特征得到所述目标类型的第三特征,包括:

将所述默认类型与所述目标类型进行比对;

在比对结果指示所述默认类型与所述目标类型不一致的情况下,将所述第二特征转换为所述目标类型的第三特征;

在比对结果指示所述默认类型与所述目标类型一致的情况下,直接将所述第二特征作

为所述目标类型的第三特征。

7. 如权利要求1所述的命令执行方法,其特征在于,在将所述目标命令定义的第一特征转换为默认类型的第二特征之后,还包括:

将所述第二特征和所述目标命令的后续操作的信息存储到内存中。

8. 如权利要求7所述的命令执行方法,其特征在于,在基于所述第二特征得到所述目标类型的第三特征之前,还包括:

在检测到所述创建命令前存在使用所述执行对象的命令且所述命令为所述目标命令的情况下,从所述内存中读取所述第二特征。

9. 如权利要求1所述的命令执行方法,其特征在于,所述基于所述目标类型的第三特征,继续执行所述目标命令的后续操作,包括:

基于所述目标类型的第三特征,对所述执行对象的特征进行调整;

基于调整后的特征,生成所述执行对象的对象文件。

10. 一种命令执行装置,其特征在于,包括:

前处理单元,被配置为在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且所述目标命令的执行对象没有被创建的情况下,将所述目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行所述目标命令的后续操作,其中,所述目标命令用于根据所述第一特征调整所述执行对象的特征;

创建单元,被配置为继续执行所述目标命令的下一条命令,直到执行所述执行对象的创建命令,基于所述创建命令创建属于目标类型的所述执行对象,并基于所述第二特征得到所述目标类型的第三特征;

后处理单元,被配置为基于所述目标类型的第三特征,继续执行所述目标命令的后续操作。

11. 如权利要求10所述的命令执行装置,其特征在于,所述前处理单元,还被配置为通过如下方式确定所述目标命令的执行对象没有被创建:确定所述目标命令的所述执行对象的类型信息;将所述类型信息指示的类型依次与执行所述UPF文件的软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定所述目标命令的所述执行对象没有被创建。

12. 如权利要求11所述的命令执行装置,其特征在于,所述前处理单元,还被配置为在确定所述目标命令的执行对象没有被创建之后,将最后与所述类型信息指示的类型比对的对象类型作为默认类型;或者,指定执行所述UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。

13. 如权利要求10所述的命令执行装置,其特征在于,所述前处理单元,还被配置为到所述目标命令且所述目标命令的执行对象已经被创建的情况下,基于所述目标命令定义的第一特征得到所述目标类型的第三特征。

14. 如权利要求13所述的命令执行装置,其特征在于,所述前处理单元,还被配置为通过如下方式确定所述目标命令的执行对象已经被创建:确定所述目标命令的所述执行对象的类型信息;将所述类型信息指示的类型依次与执行所述UPF文件的软件所支持的对象类型进行比对;在所述类型信息指示的类型与执行所述UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定所述目标命令的执行对象已经被创建,并将所述任一

个对象类型作为所述目标类型。

15. 如权利要求10所述的命令执行装置,其特征在於,所述创建单元,还被配置为将所述默认类型与所述目标类型进行比对;在比对结果指示所述默认类型与所述目标类型不一致的情况下,将所述第二特征转换为所述目标类型的第三特征;在比对结果指示所述默认类型与所述目标类型一致的情况下,直接将所述第二特征作为所述目标类型的第三特征。

16. 如权利要求10所述的命令执行装置,其特征在於,所述前处理单元,还被配置为在将所述目标命令定义的第一特征转换为默认类型的第二特征之后,将所述第二特征和所述目标命令的后续操作的信息存储到内存中。

17. 如权利要求16所述的命令执行装置,其特征在於,所述创建单元,还被配置为在基于所述第二特征得到所述目标类型的第三特征之前,在检测到所述创建命令前存在使用所述执行对象的命令且所述命令为所述目标命令的情况下,从所述内存中读取所述第二特征。

18. 如权利要求10所述的命令执行装置,其特征在於,所述后处理单元,被配置为基于所述目标类型的第三特征,对所述执行对象的特征进行调整;基于调整后的特征,生成所述执行对象的对象文件。

19. 一种电子设备,其特征在於,包括:

处理器;

用于存储所述处理器可执行指令的存储器;

其中,所述处理器被配置为执行所述指令,以实现如权利要求1至9中任一项所述的命令执行方法。

20. 一种计算机可读存储介质,其特征在於,当所述计算机可读存储介质中的指令被至少一个处理器运行时,促使所述至少一个处理器执行如权利要求1至9中任一项所述的命令执行方法。

命令执行方法及装置、电子设备、计算机可读存储介质

技术领域

[0001] 本公开涉及半导体领域,尤其涉及一种命令执行方法及装置、电子设备、计算机可读存储介质。

背景技术

[0002] 目前,实现集成电路的过程始于功能设计和验证(如使用寄存器转换级电路(Register Transfer Level,缩写为RTL)),然后进行物理设计和验证。为了方便地进行低功耗设计与验证,业界开始引入统一的电源格式(Unified Power Format,缩写为UPF),IEEE1801 标准中UPF的设计实现流程是一个真正意义统一的,被广泛采用的低功耗实现标准的流程。UPF 提供了RTL 硬件描述语言无法描述的电源设计定义,它用一些标准的语句描述用户的低功耗设计意图(Power Intent)。有了UPF的配合和支持,集成电路的设计人员可以在RTL 验证阶段就同时对RTL 代码文件和UPF 电源描述文件进行电源状态控制相关的低功耗管理策略的验证,在设计早期就可以及时发现电源管理问题,保证了设计的准确性,缩短了开发周期,减少了设计迭代。

[0003] 在UPF文件中为某些对象调整特征时,需要在明确该对象的类型之后,根据不同的类型进行相应的调整,但是创建该对象并明确其类型的命令往往在调整对象特征的命令之后定义,导致执行UPF文件的软件在执行调整对象特征的命令时,该对象还未创建,无法确定其类型,其后果是,一方面调整对象特征的命令在执行时会报错,另一方面也无法正确地对特征进行调整。但是,如果在调整某对象特征的命令之前必须先定义好该对象,则难以避免对UPF文件进行多次修改,由于UPF文件常常包含成千上万的语句,所以需要修改UPF文件中对应的大量内容,导致UPF文件开发流程加长并且不方便使用。

发明内容

[0004] 本公开提供一种命令执行方法及装置、电子设备、计算机可读存储介质,以至少解决相关技术中UPF文件开发流程加长并且不方便使用的问题。

[0005] 根据本公开实施例的第一方面,提供一种命令执行方法,包括:在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且目标命令的执行对象没有被创建的情况下,将目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行目标命令的后续操作,其中,目标命令用于根据第一特征调整执行对象的特征;继续执行目标命令的下一条命令,直到执行到执行对象的创建命令,基于创建命令创建属于目标类型的执行对象,并基于第二特征得到目标类型的第三特征;基于目标类型的第三特征,继续执行目标命令的后续操作。

[0006] 可选地,通过如下方式确定目标命令的执行对象没有被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定目标命令的执行对象没有被创建。

[0007] 可选地,在确定目标命令的执行对象没有被创建之后,还包括:将最后与类型信息

指示的类型比对的对象类型作为默认类型;或者,指定执行UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。

[0008] 可选地,若检测到目标命令且目标命令的执行对象已经被创建的情况下,基于目标命令定义的第一特征得到目标类型的第三特征;基于目标类型的第三特征,继续执行目标命令的后续操作。

[0009] 可选地,通过如下方式确定目标命令的执行对象已经被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在类型信息指示的类型与执行UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定目标命令的执行对象已经被创建,并将任一个对象类型作为目标类型。

[0010] 可选地,基于第二特征得到目标类型的第三特征,包括:将默认类型与目标类型进行比对;在比对结果指示默认类型与目标类型不一致的情况下,将第二特征转换为目标类型的第三特征;在比对结果指示默认类型与目标类型一致的情况下,直接将第二特征作为目标类型的第三特征。

[0011] 可选地,在将目标命令定义的第一特征转换为默认类型的第二特征之后,还包括:将第二特征和目标命令的后续操作的信息存储到内存中。

[0012] 可选地,在基于第二特征得到目标类型的第三特征之前,还包括:在检测到创建命令前存在使用执行对象的命令且命令为目标命令的情况下,从内存中读取第二特征。

[0013] 可选地,基于目标类型的第三特征,继续执行目标命令的后续操作,包括:基于目标类型的第三特征,对执行对象的特征进行调整;基于调整后的特征,生成执行对象的对象文件。

[0014] 根据本公开实施例的第二方面,提供一种命令执行装置,包括:前处理单元,被配置为在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且目标命令的执行对象没有被创建的情况下,将目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行目标命令的后续操作,其中,目标命令用于根据第一特征调整执行对象的特征;创建单元,被配置为继续执行目标命令的下一条命令,直到执行到执行对象的创建命令,基于创建命令创建属于目标类型的执行对象,并基于第二特征得到目标类型的第三特征;后处理单元,被配置为基于目标类型的第三特征,继续执行目标命令的后续操作。

[0015] 可选地,前处理单元,还被配置为通过如下方式确定目标命令的执行对象没有被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定目标命令的执行对象没有被创建。

[0016] 可选地,前处理单元,还被配置为在确定目标命令的执行对象没有被创建之后,将最后与类型信息指示的类型比对的对象类型作为默认类型;或者,指定执行UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。

[0017] 可选地,前处理单元,还被配置为到目标命令且目标命令的执行对象已经被创建的情况下,基于目标命令定义的第一特征得到目标类型的第三特征。

[0018] 可选地,前处理单元,还被配置为通过如下方式确定目标命令的执行对象已经被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的

软件所支持的对象类型进行比对;在类型信息指示的类型与执行UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定目标命令的执行对象已经被创建,并将任一个对象类型作为目标类型。

[0019] 可选地,创建单元,还被配置为将默认类型与目标类型进行比对;在比对结果指示默认类型与目标类型不一致的情况下,将第二特征转换为目标类型的第三特征;在比对结果指示默认类型与目标类型一致的情况下,直接将第二特征作为目标类型的第三特征。

[0020] 可选地,前处理单元,还被配置为在将目标命令定义的第一特征转换为默认类型的第二特征之后,将第二特征和目标命令的后续操作的信息存储到内存中。

[0021] 可选地,创建单元,还被配置为在基于第二特征得到目标类型的第三特征之前,在检测到创建命令前存在使用执行对象的命令且命令为目标命令的情况下,从内存中读取第二特征。

[0022] 可选地,后处理单元,被配置为基于目标类型的第三特征,对执行对象的特征进行调整;基于调整后的特征,生成执行对象的对象文件。

[0023] 根据本公开实施例的第三方面,提供了一种电子设备,包括:处理器;用于存储处理器可执行指令的存储器;其中,处理器被配置为执行指令,以实现根据本公开的命令执行方法。

[0024] 根据本公开实施例的第四方面,提供了一种计算机可读存储介质,当计算机可读存储介质中的指令被至少一个处理器运行时,促使至少一个处理器执行如上根据本公开的命令执行方法。

[0025] 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,包括计算机指令,计算机指令被处理器执行时实现根据本公开的命令执行方法。

[0026] 本公开的实施例提供的技术方案至少带来以下有益效果:

根据本公开的命令执行方法及装置、电子设备、计算机可读存储介质,在执行UPF文件中命令时,检测到目标命令且目标命令的执行对象没有被创建,可以先将该执行对象的特征转换为默认类型的第一特征,然后暂停执行目标命令的后续操作,由于在目标命令执行时,执行对象还没有被创建,此时暂停执行目标命令的后续操作,可以避免目标命令的后续操作按错误的特征调整执行对象;本公开中暂停执行目标命令的后续操作后,可以继续执行下一条命令,直到执行该执行对象的创建命令,按创建命令创建目标类型的执行对象,并基于该执行对象默认类型的第二特征得到为目标类型的第三特征,此时执行对象已经被创建,且也基于原来默认类型的第二特征得到了执行对象实际类型的第三特征,从而可以基于执行对象的目标类型的第三特征,继续执行目标命令的后续操作,使得目标命令的后续操作可以按正确的特征调整执行对象,从而通过本公开,可以在目标命令的后面创建定义执行对象,不再严格要求执行对象必须在目标命令执行前创建定义,即创建命令可以在目标命令的后面开始执行,避免了在调整某对象特征的命令之前必须先定义好该对象,而大量修改UPF文件中对应的内容,导致的开发流程加长及不方便使用的问题。因此,本公开解决了相关技术中的UPF文件开发流程加长并且不方便使用的问题。

[0027] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

附图说明

[0028] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本公开的实施例,并与说明书一起用于解释本公开的原理,并不构成对本公开的不当限定。

[0029] 图1是根据一示例性实施例示出的一种命令执行方法的流程图;
图2是根据一示例性实施例示出的一种命令的执行流程示意图;
图3是根据一示例性实施例示出的一种创建命令的执行流程示意图;
图4是根据一示例性实施例示出的另一种创建命令的执行流程示意图;
图5是根据一示例性实施例示出的一种命令执行方法的系统示意图;
图6是根据一示例性实施例示出的一种命令执行装置的框图;
图7是根据本公开实施例的一种电子设备700的框图。

具体实施方式

[0030] 为了使本领域普通人员更好地理解本公开的技术方案,下面将结合附图,对本公开实施例中的技术方案进行清楚、完整地描述。

[0031] 需要说明的是,本公开的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本公开的实施例能够以除了在这里图示或描述的那些以外的顺序实施。以下实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[0032] 在此需要说明的是,在本公开中出现的“若干项之中的至少一项”均表示包含“该若干项中的任意一项”、“该若干项中的任意多项的组合”、“该若干项的全体”这三类并列的情况。例如“包括A和B之中的至少一个”即包括如下三种并列的情况:(1)包括A;(2)包括B;(3)包括A和B。又例如“执行步骤一和步骤二之中的至少一个”,即表示如下三种并列的情况:(1)执行步骤一;(2)执行步骤二;(3)执行步骤一和步骤二。

[0033] 目前,集成电路(Integrated Circuit,缩写为IC)的设计通常使用计算机辅助设计(Computer Aided Design,缩写为CAD)工具或电子设计自动化(Electronic design automation,缩写为EDA)系统执行。为了设计集成电路,设计人员首先使用高级硬件设计语言(Hardware Description Language,缩写为HDL)创建IC器件的高级行为描述。HDL的常见示例包括Verilog HDL和超高速硬件描述语言(Very-High-Speed Hardware Description Language,缩写为VHDL)。

[0034] 以EDA系统设计集成电路(IC)为例,EDA系统通常接收使用高级硬件设计语言创建的IC器件的高级行为描述,并将这种高级行为描述转换为各种抽象级别的网表。本质上,实现集成电路的过程始于功能设计和验证(例如使用寄存器转换级电路(Register Transfer Level,缩写为RTL)),然后进行物理设计和验证。为了方便地进行低功耗设计与验证,业界开始引入统一的电源格式(Unified Power Format,缩写为UPF),IEEE1801标准中UPF的设计实现流程是一个真正意义统一的,被广泛采用的低功耗实现标准的流程。UPF提供了RTL硬件描述语言无法描述的电源设计定义,它用一些标准的语句描述用户的低功耗设计意图(Power Intent)。有了UPF的配合和支持,集成电路的设计人员可以在RTL验

证阶段,同时对RTL 代码文件和UPF 电源描述文件进行电源状态控制相关的低功耗管理策略的验证,在设计早期就可以及时发现电源管理问题,保证了设计的准确性,缩短了开发周期,减少了设计迭代。因此,在实际应用过程中,通常会在原有的RTL 设计流程基础上增加一个UPF 文件,以实现UPF与RTL的并行设计。

[0035] 然而,在UPF文件中命令顺序执行的过程中,对于调整对象特征的命令,常常需要在该命令执行前先定义好该对象,以便该命令根据定义的对象类型正确执行,若定义该对象的相关命令在该命令执行之后,则在执行该命令时会产生错误。但是,由于UPF文件常常包含成千上万的语句,如果在调整某对象特征的命令之前必须先定义好该对象,则需要修改UPF文件中对应的大量内容,导致UPF文件开发流程加长并且不方便使用。

[0036] 下面以调整对象特征的命令是add_power_state命令为例简要上述问题,其中,add_power_state命令的语法规则如下:

```
add_power_state
[-supply|-domain | -group | -model | -instance]object_name
[-update]
[-state {state_name
[-logic_expr {boolean_expression}]
[-supply_expr {boolean_expression}]
[-power_expr {power_expression}]
[-simstatesimstate]
[-legal|-illegal]
}]]*
[-complete]
```

在上述语法规则中,[-supply | -domain | -group | -model | -instance]表示该对象的类型(object type),其不属于必须写入命令语句中的内容,在命令语句中常被省略。为简化描述,本公开仅讨论supply、domain和group这三种类型,其中,supply_set表示供电电源集合,group表示电源状态表集,domain表示电源区域。

[0037] 实际应用中,若针对某个对象(object)执行add_power_state操作,在该对象的类型(type)分别是supply、group、domain的情况下,add_power_state命令的行为和结果均不一致。例如,在上述语法规则中,-supply_expr和-simstate仅适用于对象类型为supply的情况。因此,在UPF文件中,需要在使用add_power_state命令前预先定义好该命令的执行对象,以根据执行对象的类型添加正确的低功耗特征,若定义执行对象的命令在add_power_state语句之后,则会导致add_power_state命令在执行时无法确定执行对象的类型,使得难以正确执行add_power_state的动作,例如,某个对象被定义为supply_set时,其对象的类型为supply,却为其添加了domain类型的特征,那么add_power_state命令在执行时就会产生错误。而如果在使用add_power_state命令前必须先定义好其中的执行对象,则难以避免对UPF文件进行多次修改,由于UPF文本中有成千上万的语句,所以需要大量的修改对应的文本内容(牵一发而动全身的意思),使得UPF开发流程加长并且不方便使用。

[0038] 针对上述问题,本公开提供了一种命令执行方法及装置、电子设备、计算机可读存储介质,在执行UPF文件中命令时,检测到目标命令且目标命令的执行对象没有被创建,可

以先将目标命令定义的第一特征转换为默认类型的第二特征,然后暂停执行目标命令的后续操作,继续执行下一条命令,直到执行该执行对象的创建命令,按创建命令创建目标类型的执行对象,并基于第二特征得到目标类型的第三特征,从而可以基于目标类型的第三特征,继续执行目标命令的后续操作,使得目标命令的执行对象可以在目标命令执行后进行创建定义,无需一定在目标命令执行前创建定义,避免了在目标命令执行前创建定义所需的执行对象,大量修改UPF文件中对应的内容,能够缩短UPF文件的开发流程。

[0039] 下面,将参照附图详细描述根据本公开的示例性实施例的命令执行方法及装置。

[0040] 图1是根据一示例性实施例示出的一种命令执行方法的流程图,如图1所示,命令执行方法包括以下步骤:

在步骤S101中,在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且目标命令的执行对象没有被创建的情况下,将目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行目标命令的后续操作,其中,目标命令用于根据第一特征调整执行对象的特征。

[0041] 上述目标命令是用于根据第一特征调整执行对象的特征的命令,如目标命令可以根据第一特征添加执行对象的特征,也可以根据第一特征删除执行对象的特征,还可以根据第一特征修正执行对象的特征,对此本公开并不进行限定。

[0042] 作为示例,以目标命令用于根据第一特征添加执行对象的特征为例,目标命令可以是add_power_state命令,也可以是与add_power_state命令形式类似的其他命令,对此本公开并不进行限定。

[0043] 上述目标命令定义的第一特征,根据不同的命令形式会有不同的表达形式,下面以add_power_state命令为例进行简要说明,在该add_power_state命令的语法规则中,object_name即执行对象的名称;state_name即目标命令定义的第一特征的名称或者简化表示,如果一个执行对象的特征调整需要依据多个特征,该命令对应可以表示为add_power_state objectA -state {A1, A2, ..., Am};其中,A1、A2、...、Am即调整执行对象objectA所依据的多个特征的名称或者简化表示,就该命令本身而言,A1、A2、...、Am是该命令中定义的需要添加至执行对象objectA的多个特征的简化表示。

[0044] 上述UPF文件可以通过CAD软件执行,也可以通过EDA软件执行,还可以通过其他可以执行UPF文件的软件执行,对此本公开并不进行限定。

[0045] 作为示例,以EDA软件执行UPF文件,目标命令为add_power_state命令为例,首先,EDA软件载入用户提供的设计文件,建立UPF文件的服务对象,如通过如下命令实现:read_design user_desgins,对此本公开并不进行限定;其次,EDA软件读入用户提供的针对已加载的设计文件的UPF文件,UPF文件部分内容示例如下:

```
add_power_state objectA -state {A1, A2, ..., Am}
add_power_state objectB -state {B1, B2, ..., Bn}
create_power_domain objectA
create_supply_set objectB
```

由于UPF文件中命令执行到add_power_state objectA -state {A1, A2, ..., Am}命令时,命令中的执行对象objectA还未被创建,此时可以先执行该命令的前操作,并暂停执行该命令的后操作(即上述步骤中的后续操作);同理,由于UPF文件中命令执行到add_

power_state object B -state{B1, B2, ..., Bn} 命令时,命令中的执行对象objectB还未被创建,此时也可以先执行该命令的前操作,并暂停执行该命令的后操作(即上述步骤中的后续操作)。值得说明的是,上述命令中的(A1, A2, ..., Am)、(B1, B2, ..., Bn)仅用于示例,以表示调整执行对象所依据的多个特征的名称或者简化表示,对此本公开并不进行限定。

[0046] 为了方便理解上述实施例中的前操作和后操作,下面通过add_power_state命令的执行流程,对前操作和后操作进行简单介绍,以add_power_state A -state{A1, A2}命令为例,假设命令中的执行对象A在该命令前已经被建立,此时该命令的执行流程如图2所示,操作1(action 1)为前操作,操作2(action 2)、操作3(action 3)和操作4(action 4)为后操作,其中,前操作和后操作分别介绍如下:

前操作action 1包括如下步骤:

2.1、对add_power_state A -state{A1, A2}命令进行解析,确定命令中的执行对象A的特征中需要添加的表达特征A1、A2。

[0047] 2.2、判断执行对象A的对象类型(object type):可以先判断执行对象A的对象类型是否为domain,再判断A的对象类型是否为supply_set,如果都不是,则执行对象A的对象类型为group;如果此时执行对象A还没有被创建,则无法正常确定出执行对象A的对象类型,经过上述判断过程后,可以默认A的对象类型为group。值得说明的是,上述对象类型的判断顺序仅为示例性描述,具体应用时可根据实际需求进行调整,本公开不进行限定。

[0048] 2.3、对A1、A2进行转换:如果执行对象A的对象类型为domain,将A1、A2转换为domain类型的正确表达,转换后得到的特征表示为A1_D、A2_D;如果执行对象A的对象类型为supply,将A1、A2转换为supply类型的正确表达,转换后得到的特征表示为A1_S、A2_S;如果执行对象A的对象类型为group,将A1、A2转换为group类型的正确表达,转换后得到的特征表示为A1_G、A2_G。

[0049] 后操作action 2、action 3和action 4包括如下步骤:

2.4、为执行对象A添加低功耗特征,生成相应类型(type)的obj文件(即扩展名为obj的对象文件):

action2:基于转换后的表达特征A1_D、A2_D,生成相应的obj文件,该obj文件的执行对象名称(object_name)为A,对象类型为domain(为方便区分描述,图2中使用object A_D表示生成的obj文件);

action3:基于转换后的表达特征A1_S、A2_S,生成相应的obj文件,该obj文件的执行对象名称(object_name)为A,对象类型为supply(为方便区分描述,图2中使用object A_S表示生成的obj文件);

action4:基于转换后的表达特征A1_G、A2_G,生成相应的obj文件,该obj文件的执行对象名称(object_name)为A,对象类型为group(为方便区分描述,图2中使用object A_G表示生成的obj文件)。

[0050] 根据本公开的示例性实施例,在将目标命令定义的第一特征转换为默认类型的第二特征之后,还可以将第二特征和目标命令后续操作的信息存储到内存中。根据本实施例,将转换后的默认类型的第二特征以及目标命令后续操作的信息存储到内存中,方便后续检测到执行对象的创建命令后,快速调用所需的信息,以执行后续操作。

[0051] 当然,上述默认类型的第二特征和目标命令后续操作的信息也可以存储在其他地

方,如硬盘等,对此本公开并不进行限定。上述目标命令后续操作的信息可以是目标命令后续操作程序语句本身,也可以是其他信息,对此本公开并不进行限定。如果将后续操作程序语句本身作为相关信息进行存储,能够快速实现目标命令后续操作的信息的存储,即无需在进行相应的转化、提取等操作,后续在调用该信息时,也可以直接调用使用,便于使用。

[0052] 根据本公开的示例性实施例,可以通过如下方式确定目标命令的执行对象没有被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定目标命令的执行对象没有被创建。根据本实施例,通过依次与执行UPF文件的软件所支持的对象类型进行比对,可以快速的知晓目标命令的执行对象是否创建,因为如果目标命令的执行对象被创建,必然与执行UPF文件的软件所支持的对象类型中的一个类型一致。

[0053] 上述执行UPF文件的软件可以是CAD软件,也可以是EDA软件,还可以是其他可以执行UPF文件的软件,对此本公开并不进行限定。例如,以EDA软件执行UPF文件为例,确定目标命令的执行对象的类型信息后,可以将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定目标命令的执行对象没有被创建。

[0054] 对于上述类型信息,如果执行对象已经被创建,则确定的类型信息所指示的对象类型就必然是相应软件执行目标命令所支持的对象类型中的一种;如果执行对象尚未创建,则确定的类型信息可能为空或者0值等,具体取决于实际情况,其指示的类型可以是一个特殊的类型,对此本公开并不进行限定。

[0055] 作为示例,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,可以先确定目标命令的执行对象的类型信息,然后,将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对,如先判断该目标命令中的执行对象A的类型信息所指示的对象类型是否为domain,再判断执行对象A的类型信息所指示的对象类型是否为supply_set,然后判断执行对象A的类型信息所指示的对象类型是否为group;如果判断结果均指示类型不一致的情况下,则无法正常确定出执行对象A的对象类型,即经过上述判断过程后确定执行对象A的对象类型不是上述三种类型时,确定执行对象A没有被创建。值得说明的是,本实施例的对象类型的判断顺序仅为示例性描述,具体应用时可根据实际需求进行调整,本公开也不进行限定。

[0056] 根据本公开的示例性实施例,在确定目标命令的执行对象没有被创建之后,还可以将最后与类型信息指示的类型比对的对象类型作为默认类型;或者,指定执行UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。根据本公开的实施例,可以将最后与类型信息指示的类型比对的对象类型作为默认类型,也可以指定一个对象类型为默认类型,提高了默认类型的灵活度,从而可以根据需要确定用户所需的默认类型。

[0057] 作为示例,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,如果将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对后,确定执行对象A的对象类型不是上述三种类型时,可以指定上述三种类型中的一种类型为默认类型,如group,并将A1、A2转换为默认类型(group类型)的正确表达,转换后得到的特征表示为A1_G、A2_G,对于默认类型指定的是哪一种类型,本公开并不进行限定;也可以将最后与类型信息指示的类型比对的对象类型作为默认类型,对此本公开并不进行限

定。

[0058] 根据本公开的示例性实施例,若检测到目标命令且目标命令的执行对象已经被创建的情况下,基于目标命令定义的第一特征得到目标类型的第三特征;基于目标类型的第三特征,继续执行目标命令的后续操作。根据本实施例,如果目标命令的执行对象已经被创建,可以基于目标命令定义的第一特征直接得到目标类型的第三特征,无需经过默认类型的转换过程,提高了命令执行的效率。

[0059] 作为示例,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,如果执行对象A已经被预先创建完成且执行对象A的对象类型就是group的情况下,也即执行对象A的对象类型确实是group(即目标类型),此时可以将目标命令中定义的第一特征直接转换为group类型的第三特征,并进行目标命令的后操作即可。

[0060] 根据本公开的示例性实施例,通过如下方式确定目标命令的执行对象已经被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在类型信息指示的类型与执行UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定目标命令的执行对象已经被创建,并将任一个对象类型作为目标类型。根据本实施例,通过依次与执行UPF文件的软件所支持的类型进行比对,可以快速的知晓目标命令的执行对象是否创建,因为如果目标命令的执行对象被创建,必然与执行UPF文件的软件所支持的类型中的一个类型一致。

[0061] 作为示例,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,如果将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对后,确定执行对象A的对象类型属于三种类型中的一种类型,则说明执行对象A在目标命令执行前已经被创建,将执行UPF文件的软件所支持的对象类型中与类型信息指示的类型一致的对象类型,确定为目标类型。

[0062] 返回图1,在步骤S102中,继续执行目标命令的下一条命令,直到执行到执行对象的创建命令,基于创建命令创建属于目标类型的执行对象,并基于第二特征得到目标类型的第三特征。

[0063] 具体地,上述继续执行目标命令的下一条命令指的是,跳过目标命令的后操作,直接执行目标命令的下一条命令,并继续执行UPF文件中后续的命令,直至执行到目标命令中执行对象的创建命令,此时开始创建该执行对象,在创建完目标命令中执行对象后,则可以将默认类型的第二特征转换为目标类型的特征,并继续执行目标命令的后续操作。

[0064] 例如,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,假设继续执行UPF文件中目标命令的后续命令时,执行到`create_power_domain A`命令时,会马上进行执行对象A的创建操作以及相应的转换操作,具体执行流程如图3所示,主要包含如下步骤:

3.1、对`create_power_domain A`命令进行解析;

3.2、判断执行对象A的对象类型:可以先判断执行对象A的对象类型是否为domain,再判断执行对象A的对象类型是否为supply_set,以及判断执行对象A的对象类型是否为group;由于执行对象A已经被定义为domain,所以此时能够明确的确定执行对象A的对象类型为domain;

3.3、收集(collect)上述储存的默认类型的特征A1_G、A2_G:在目标命令`add_`

power_state A -state {A1, A2} 的前操作中,特征A1、A2已经默认转换为A1_G、A2_G,此时可以将转换后的A1_G、A2_G从内存中读取出来并收集起来;

3.4、对特征A1_G、A2_G进行转换:将收集到的特征A1_G、A2_G转换为domain类型的正确表达,转换后得到的特征表示为A1_D、A2_D。

[0065] 又例如,仍以EDA软件执行UPF文件,目标命令为add_power_state A -state {A1, A2} 命令为例,假设继续执行UPF文件中目标命令的后续命令时,执行到create_supply_set A命令时,会马上进行执行对象A的创建操作以及相应的转换操作,具体执行流程如图4所示,主要包含如下步骤:

4.1、对create_supply_set A命令进行解析;

4.2、判断执行对象A的对象类型:可以先判断执行对象A的对象类型是否为domain,再判断执行对象A的对象类型是否为supply_set,以及判断执行对象A的对象类型是否为group;由于执行对象A已经被定义为supply_set,所以此时能够明确的确定执行对象A的对象类型为supply;

4.3、收集上述储存的默认类型的特征A1_G、A2_G:在目标命令add_power_state A -state {A1, A2} 的前操作中,特征A1、A2默认转换为A1_G、A2_G,此时可以将转换后的A1_G、A2_G从内存中读取出来并收集起来;

4.4、对特征A1_G、A2_G进行转换:将收集到的特征A1_G、A2_G转换为supply类型的正确表达,转换后得到的特征表示为A1_S、A2_S。

[0066] 因此,本公开在执行对象A还没有被创建的情况下,目标命令add_power_state A -state {A1, A2} 的前操作1(action 1)可以默认执行对象A的对象类型为group,从而默认将特征A1、A2转换为A1_G、A2_G;而在执行对象A创建结束后,可以将默认类型的特征A1、A2转换到对应的目标类型的正确表达,如A1_D、A2_D和A1_S、A2_S。因此,无论执行对象A的创建命令是在目标命令add_power_state A -state {A1, A2} 之前还是之后,UPF文件都会正确执行所包含的命令,不会产生冲突,也无需进行额外的解析、判断、收集和转换等操作。

[0067] 根据本公开的示例性实施例,在基于第二特征得到目标类型的第三特征之前,还可以在检测到创建命令前存在使用执行对象的命令且命令为目标命令中的情况下,从内存中读取默认类型的第二特征。根据本实施例,只有在创建命令的前面的某个命令中使用了该执行对象,如目标命令使用了该执行对象,说明前面有命令在执行对象未创建前就使用该执行对象,也即有命令运行到了一半,此时才从内存中读取目标命令的相关信息进行处理,避免无意义的读取内存。

[0068] 具体地,在UPF文件中命令顺序执行的过程中,有可能先执行到执行对象的创建命令,也即在创建执行对象的命令前,并未出现使用了执行对象的命令,此时则无需从内存中读取默认类型的第二特征,因此可以避免无意义的读取内容,而如果在创建执行对象前出现了使用了执行对象的命令,如目标命令,则此时可以从内存中读取默认类型的第二特征,并进行后续处理。

[0069] 例如,上述步骤3.1对create_power_domain A进行解析和步骤4.1对create_supply_set A进行解析后,还可以进一步增加一个确定步骤,即确定创建命令的前置add_power_state相关命令预先使用了执行对象“A”,然后再判断A的对象类型,以及收集上述默认类型的特征A1_G、A2_G,也即从内存中读取默认类型的特征,否则仅需要判断执行对象A

的对象类型即可,无需从内存中读取默认类型的特征,因为此时创建执行对象A的命令前面没有出现使用了执行对象A的命令,也就不存在默认类型的特征。

[0070] 根据本公开的示例性实施例,基于第二特征得到目标类型的第三特征,可以包括但不限于:将默认类型与目标类型进行比对;在比对结果指示默认类型与目标类型不一致的情况下,将第二特征转换为目标类型的第三特征;在比对结果指示默认类型与目标类型一致的情况下,直接将第二特征作为目标类型的第三特征。根据本实施例,在目标类型与默认类型不一致时,才将默认类型的第二特征转换为目标类型的第三特征,如果目标类型与默认类型一致时,则直接根据第二特征继续执行目标命令的后续操作即可,无需再次转换特征,避免目标类型与默认类型一致时,重复转换操作带来的时间和资源的浪费。

[0071] 例如,仍以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,假设默认类型的特征是A1_G、A2_G,遇到的创建命令是`create_power_group A`,也即创建的执行对象A的目标类型是group,此时说明目标类型和默认类型是一样的,则无需将默认类型的特征转换为目标类型的特征,即可以直接采用默认类型的特征A1_G、A2_G,继续执行目标命令的后操作;如果遇到的创建命令是`create_power_domain A`,也即创建的执行对象A的目标类型是domain,说明目标类型和默认类型是不一样的,则需要将默认类型的特征转换为目标类型的特征,即此时需要将默认类型的特征A1_G、A2_G转换为domain类型的正确表达,转换后得到的特征表示为A1_D、A2_D,此时根据特征A1_D、A2_D,继续执行目标命令的后操作。

[0072] 返回图1,在步骤S103中,基于目标类型的第三特征,继续执行目标命令的后续操作。

[0073] 根据本公开的示例性实施例,基于目标类型的第三特征,继续执行目标命令的后续操作,可以包括:基于目标类型的第三特征,对执行对象的特征进行调整;基于调整后的特征,生成执行对象的对象文件。根据本实施例,根据目标类型的第三特征,可以生成执行对象正确的对象文件。

[0074] 具体地,上述目标命令的后续操作可以是图2所示后操作,即为A添加低功耗特征,生成相应类型(type)的obj文件,对此本公开并不进行限定。

[0075] 为了更好的理解上述实施例,下面结合图5进行系统的说明,图5是根据一示例性实施例示出的一种命令执行方法的系统示意图,如图5所示,以EDA软件执行UPF文件,目标命令为`add_power_state A -state {A1, A2}`命令为例,目标命令`add_power_state A -state {A1, A2}`执行完前操作后(前操作在前面已经详细论述,此处不在展开论述),可以先确定目标命令的执行对象的类型信息,然后,将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对,如判断目标命令中执行对象A的类型信息所指示的对象类型是否为domain,再判断执行对象A的类型信息所指示的对象类型是否为supply_set;若此时执行对象A还没有被创建,则无法正常确定出执行对象A的对象类型,即经过上述判断过程后确定执行对象A的对象类型不是上述二种类型,此时可以默认执行对象A的对象类型为一种类型,如group,并将特征A1、A2转换为默认类型(group类型)的正确表达,转换后得到的特征表示为A1_G、A2_G。需要说明的是,为了避免出现执行对象A已经被创建且执行对象A的对象类型就是group的情况下,也即执行对象A的对象类型明确确定是group的情况下,还可以进一步的判断执行对象A的类型信息所指示的对象类型是否为group,如果也不是group的情

况下,再默认执行对象A的对象类型为一种类型,如group。在转换为默认类型的特征后,可以将默认类型的特征以及后操作的相关信息进行存储,直到执行到执行对象A的创建命令,如图5所示的两个示例create_power_domain A或者create_supply_set A,其中,create_power_domain会在action 2之前执行,create_supply_set会在action 3之前执行,具体地,在创建命令为create_power_domain A时,执行流程如图3所示,在创建命令为create_supply_set A时,执行流程如图4所示,由于前面已经详细论述,此处不在展开论述。在得到目标类型的特征后,就可以继续目标命令的后操作了。在图5所示总体执行框架下,即便是在add_power_state命令之后定义执行对象A,也能够为不同类型的执行对象各自正确添加低功耗特征。

[0076] 综上,在UPF文件中为某些执行对象添加低功耗特征时,需要明确这些执行对象的类型才能正确为其添加低功耗特征,但是创建该执行对象并明确对象类型的动作(例如create_supply_set、create_power_domain、create_power_state_group等)如果是在add_power_state操作之后,例如,UPF文件中add_power_state objectA操作需要明确objectA的对象类型,而create_power_domain objectA虽然明确定义了执行对象A的对象类型为domain,但是却在add_power_state命令之后定义的,这样UPF文件就会出现执行错误。而为了避免执行出现错误,本公开将add_power_state命令划分为前操作和后操作,在add_power_state命令的前操作执行后,不会立即执行add_power_state的后操作,而是会暂停执行add_power_state的后操作,并在定义了add_power_state所使用的执行对象,即运行完创建该执行对象的命令后,再执行add_power_state命令的后操作,从而可以生成正确添加了低功耗特征的obj文件。

[0077] 通过本公开,在执行add_power_state等形式的命令时,即使是在该命令后面定义的所使用的执行对象,即没有提前定义好该命令所使用的执行对象,也可以正确执行add_power_state等形式的命令,即可以为不同类型的执行对象执行各自正确的添加低功耗特征的行为。

[0078] 本公开在完成UPF文件和设计信息的载入后,则可以根据UPF设计标准检查设计的缺陷,包括但并不限于:1) 载入的设计信息中存在导致电信号状态处于无法控制的异常状态,如供能电源域OFF- \gt ;ON的设计错误,2) 载入的设计信息中存在无用的用电功耗,3) 载入的设计信息中存在电压不同区域有连接,但是不存在电压转换装置,当然还可以包括其他设计缺陷,对此本公开并不进行限定。

[0079] 图6是根据一示例性实施例示出的一种命令执行装置的框图。参照图6,该装置包括前处理单元60、创建单元62和后处理单元64。

[0080] 前处理单元60,被配置为在顺序执行UPF文件中的多条命令的过程中,若检测到目标命令且目标命令的执行对象没有被创建的情况下,将目标命令定义的第一特征转换为默认类型的第二特征,并暂停执行目标命令的后续操作,其中,目标命令用于根据第一特征调整执行对象的特征;创建单元62,被配置为继续执行目标命令的下一条命令,直到执行执行对象的创建命令,基于创建命令创建属于目标类型的执行对象,并基于第二特征得到目标类型的第三特征;后处理单元64,被配置为基于目标类型的第三特征,继续执行目标命令的后续操作。

[0081] 上述目标命令是用于根据第一特征调整执行对象的特征的命令,如目标命令可以

根据第一特征添加执行对象的特征,也可以根据第一特征删除执行对象的特征,还可以根据第一特征修正执行对象的特征,对此本公开并不进行限定。

[0082] 作为示例,以目标命令用于根据第一特征添加执行对象的特征为例,目标命令可以是add_power_state命令,也可以是与add_power_state命令形式类似的其他命令,对此本公开并不进行限定。

[0083] 上述目标命令定义的第一特征,根据不同的命令形式会有不同的表达形式,下面以add_power_state命令为例进行简要说明,在该add_power_state命令的语法规则中,object_name即执行对象的名称;state_name即目标命令定义的第一特征的名称或者简化表示,如果一个执行对象的特征调整需要依据多个特征,该命令对应可以表示为add_power_state objectA -state {A1, A2, ..., Am};其中,A1、A2、...、Am即调整执行对象objectA所依据的多个特征的名称或者简化表示,就该命令本身而言,A1、A2、...、Am是该命令中定义的需要添加至执行对象objectA的多个特征的简化表示。

[0084] 上述继续执行目标命令的下一条命令指的是,跳过目标命令的后操作,直接执行目标命令的下一条命令,并继续执行UPF文件中后续的命令,直至执行到目标命令中执行对象的创建命令,此时开始创建该执行对象,在创建完目标命令中执行对象后,则可以将默认类型的第二特征转换为目标类型的特征,并继续执行目标命令的后续操作。

[0085] 例如,仍以EDA软件执行UPF文件,目标命令为add_power_state A -state {A1, A2} 命令为例,可以先确定目标命令的执行对象的类型信息,然后,将类型信息指示的类型依次与EDA软件所支持的对象类型进行比对,如判断该目标命令中执行对象A的类型信息所指示的对象类型是否为domain,再判断执行对象A的类型信息所指示的对象类型是否为supply_set,然后判断执行对象A的类型信息所指示的对象类型是否为group;如果此时执行对象A还没有被创建,则无法正常确定出执行对象A的对象类型,即经过上述判断过程后确定执行对象A的对象类型不是上述三种类型时,此时可以默认A的对象类型为上述三种类型中的一种类型,如group,并将A1、A2转换为默认类型(group类型)的正确表达,转换后得到的特征表示为A1_G、A2_G,对于默认类型选择是哪一种本公开并不进行限定。值得说明的是,本实施例的对象类型的判断顺序仅为示例性描述,具体应用时可根据实际需求进行调整,本公开也不进行限定。

[0086] 如果判断过程确定执行对象A的对象类型属于三种类型中的一种类型,则说明执行对象A在目标命令执行前已经被创建,如果执行对象A已经被预先创建完成且执行对象A的对象类型就是group的情况下,也即执行对象A的对象类型确实是group(即目标类型),此时可以将目标命令中定义的第一特征直接转换为group类型的第三特征,并进行目标命令的后操作即可。

[0087] 假设继续执行UPF文件中目标命令的后续命令时,执行到create_power_domain A命令时,会马上进行执行对象A的创建操作以及相应的转换操作,具体如图3所示,此处不在展开论述,在转化为目标类型的特征,并继续执行目标命令的后操作。

[0088] 根据本公开的实施例,前处理单元60,还被配置为在将目标命令定义的第一特征转换为默认类型的第二特征之后,将第二特征和目标命令的后续操作的信息存储到内存中。根据本实施例,将转换后的默认类型的特征以及目标命令后续操作的信息存储到内存中,方便后续检测到执行对象的创建命令后,快速调用所需的信息,以执行后续操作。

[0089] 根据本公开的实施例,创建单元62,还被配置为在基于第二特征得到目标类型的第三特征之前,在检测到创建命令前存在使用执行对象的命令且命令为目标命令的情况下,从内存中读取第二特征。根据本实施例,只有在创建命令的前面的某个命令中使用了该执行对象,如目标命令使用了该执行对象,说明前面有命令在执行对象未创建前就使用该执行对象,也即有命令运行到了一半,此时才从内存中读取目标命令的相关信息进行处理,避免无意义的读取内存。

[0090] 根据本公开的实施例,前处理单元60,还被配置为通过如下方式确定目标命令的执行对象没有被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在比对结果均指示类型不一致的情况下,确定目标命令的执行对象没有被创建。根据本实施例,通过依次与执行UPF文件的软件所支持的对象类型进行比对,可以快速的知晓目标命令的执行对象是否创建,因为如果目标命令的执行对象被创建,必然与执行UPF文件的软件所支持的对象类型中的一个类型一致。

[0091] 根据本公开的实施例,前处理单元60,还被配置为在确定目标命令的执行对象没有被创建之后,将最后与类型信息指示的类型比对的对象类型作为默认类型;或者,指定执行UPF文件的软件所支持的对象类型中的一种对象类型作为默认类型。根据本公开的实施例,可以将最后与类型信息指示的类型比对的对象类型作为默认类型,也可以指定一个对象类型为默认类型,提高了默认类型的灵活性,从而可以根据需要确定用户所需的默认类型。

[0092] 根据本公开的实施例,前处理单元60,还被配置为到目标命令且目标命令的执行对象已经被创建的情况下,基于目标命令定义的第一特征得到目标类型的第三特征。根据本实施例,如果目标命令的执行对象已经被创建,可以基于目标命令定义的第一特征直接得到目标类型的第三特征,无需经过默认类型的转换过程,提高了命令执行的效率。

[0093] 根据本公开的实施例,前处理单元60,还被配置为通过如下方式确定目标命令的执行对象已经被创建:确定目标命令的执行对象的类型信息;将类型信息指示的类型依次与执行UPF文件的软件所支持的对象类型进行比对;在类型信息指示的类型与执行UPF文件的软件所支持的对象类型中任一个对象类型一致的情况下,确定目标命令的执行对象已经被创建,并将任一个对象类型作为目标类型。根据本实施例,通过依次与执行UPF文件的软件所支持的对象类型进行比对,可以快速的知晓目标命令的执行对象是否创建,因为如果目标命令的执行对象被创建,必然与执行UPF文件的软件所支持的对象类型中的一个类型一致。

[0094] 根据本公开的实施例,创建单元62,还被配置为将默认类型与目标类型进行比对;在比对结果指示默认类型与目标类型不一致的情况下,将第二特征转换为目标类型的第三特征;在比对结果指示默认类型与目标类型一致的情况下,直接将第二特征作为目标类型的第三特征。根据本实施例,在目标类型与默认类型不一致时,才将执行对象的默认类型的特征转换为目标类型的特征,避免目标类型与默认类型一致时,重复转换操作带来的时间和资源的浪费。

[0095] 根据本公开的实施例,后处理单元64,还被配置为基于目标类型的第三特征,对执行对象的特征进行调整;基于调整后的特征,生成执行对象的对象文件。

[0096] 根据本公开的实施例,可提供一种电子设备。图7是根据本公开实施例的一种电子设备700的框图,该电子设备包括至少一个存储器701和至少一个处理器702,所述至少一个

存储器中存储有计算机可执行指令集合,当计算机可执行指令集合被至少一个处理器执行时,执行根据本公开实施例的命令执行方法。

[0097] 作为示例,电子设备700可以是PC计算机、平板装置、个人数字助理、智能手机、或其他能够执行上述指令集合的装置。这里,电子设备1000并非必须是单个的电子设备,还可以是任何能够单独或联合执行上述指令(或指令集)的装置或电路的集合体。电子设备700还可以是集成控制系统或系统管理器的一部分,或者可被配置为与本地或远程(例如,经由无线传输)以接口互联的便携式电子设备。

[0098] 在电子设备700中,处理器702可包括中央处理器(CPU)、图形处理器(GPU)、可编程逻辑装置、专用处理器系统、微控制器或微处理器。作为示例而非限制,处理器702还可包括模拟处理器、数字处理器、微处理器、多核处理器、处理器阵列、网络处理器等。

[0099] 处理器702可运行存储在存储器中的指令或代码,其中,存储器701还可以存储数据。指令和数据还可经由网络接口装置而通过网络被发送和接收,其中,网络接口装置可采用任何已知的传输协议。

[0100] 存储器701可与处理器702集成为一体,例如,将RAM或闪存布置在集成电路微处理器等之内。此外,存储器701可包括独立的装置,诸如,外部盘驱动、存储阵列或任何数据库系统可使用的其他存储装置。存储器701和处理器702可在操作上进行耦合,或者可例如通过I/O端口、网络连接等互相通信,使得处理器702能够读取存储在存储器701中的文件。

[0101] 此外,电子设备700还可包括视频显示器(诸如,液晶显示器)和用户交互接口(诸如,键盘、鼠标、触摸输入装置等)。电子设备的所有组件可经由总线和/或网络而彼此连接。

[0102] 根据本公开的实施例,还可提供一种计算机可读存储介质,其中,当计算机可读存储介质中的指令被至少一个处理器运行时,促使至少一个处理器执行本公开实施例的命令执行方法。这里的计算机可读存储介质的示例包括:只读存储器(ROM)、随机存取可编程只读存储器(PROM)、电可擦除可编程只读存储器(EEPROM)、随机存取存储器(RAM)、动态随机存取存储器(DRAM)、静态随机存取存储器(SRAM)、闪存、非易失性存储器、CD-ROM、CD-R、CD+R、CD-RW、CD+RW、DVD-ROM、DVD-R、DVD+R、DVD-RW、DVD+RW、DVD-RAM、BD-ROM、BD-R、BD-R LTH、BD-RE、蓝光或光盘存储器、硬盘驱动器(HDD)、固态硬盘(SSD)、卡式存储器(诸如,多媒体卡、安全数字(SD)卡或极速数字(XD)卡)、磁带、软盘、磁光数据存储装置、光学数据存储装置、硬盘、固态盘以及任何其他装置,所述任何其他装置被配置为以非暂时性方式存储计算机程序以及任何相关联的数据、数据文件和数据结构并将所述计算机程序以及任何相关联的数据、数据文件和数据结构提供给处理器或计算机使得处理器或计算机能执行所述计算机程序。上述计算机可读存储介质中的计算机程序可在诸如客户端、主机、代理装置、服务器等计算机设备中部署的环境中运行,此外,在一个示例中,计算机程序以及任何相关联的数据、数据文件和数据结构分布在联网的计算机系统上,使得计算机程序以及任何相关联的数据、数据文件和数据结构通过一个或多个处理器或计算机以分布式方式存储、访问和执行。

[0103] 根据本公开实施例,提供了一种计算机程序产品,包括计算机指令,计算机指令被处理器执行时实现本公开实施例的命令执行方法。

[0104] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本公开的其他实施方案。本公开旨在涵盖本公开的任何变型、用途或者适应性变化,这些变型、用途或

者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本公开的真正范围和精神由下面的权利要求指出。

[0105] 应当理解的是，本公开并不局限于上面已经描述并在附图中示出的精确结构，并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。

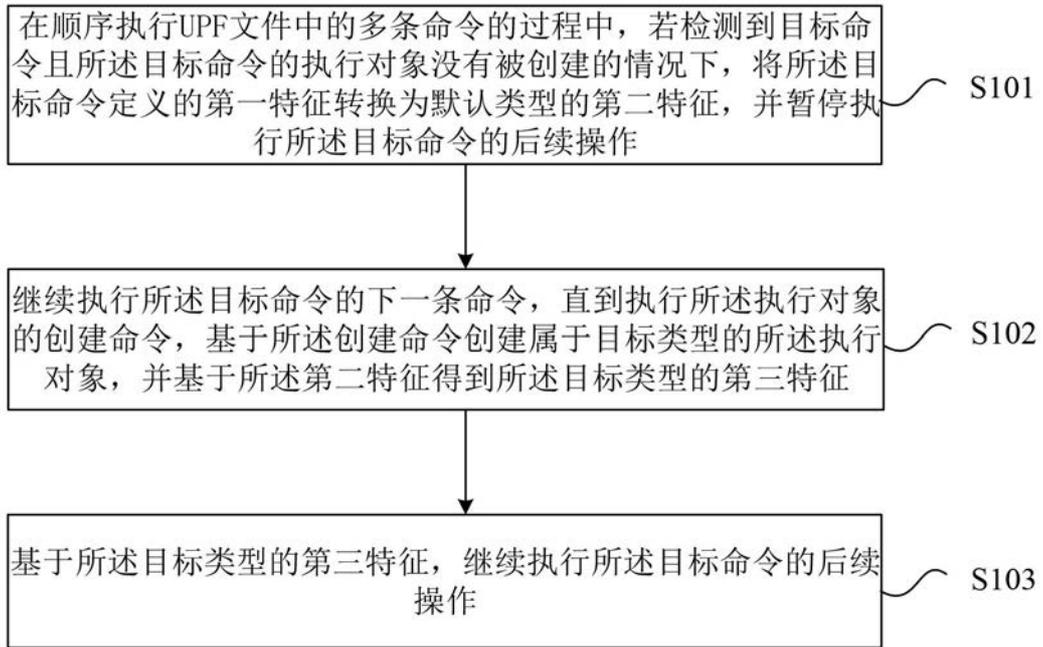


图1

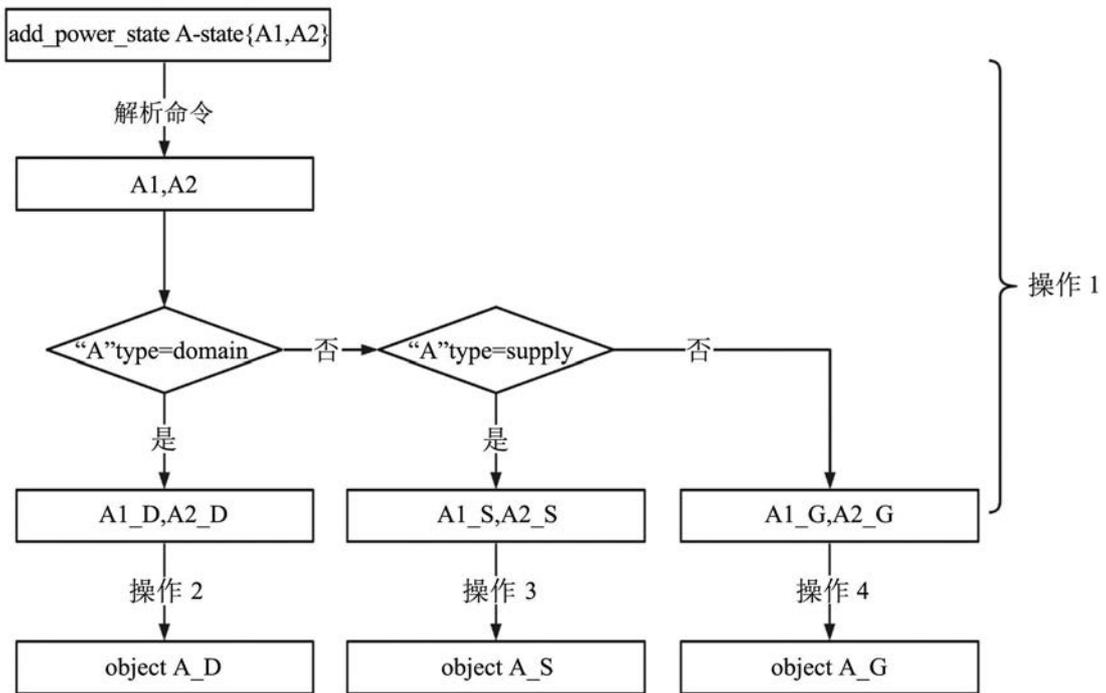


图2

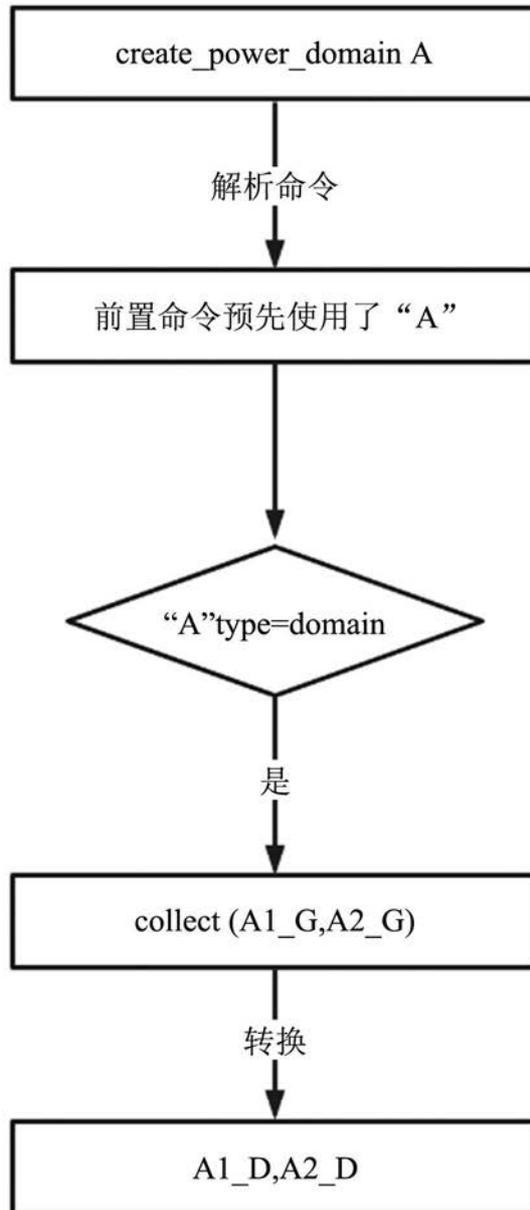


图3

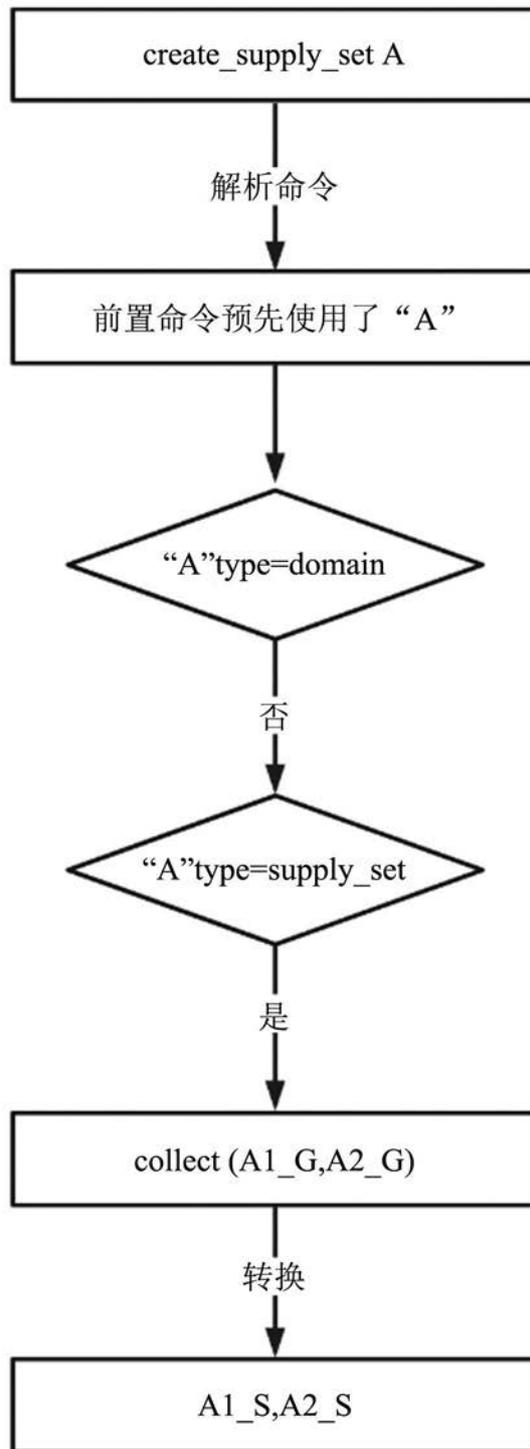


图4

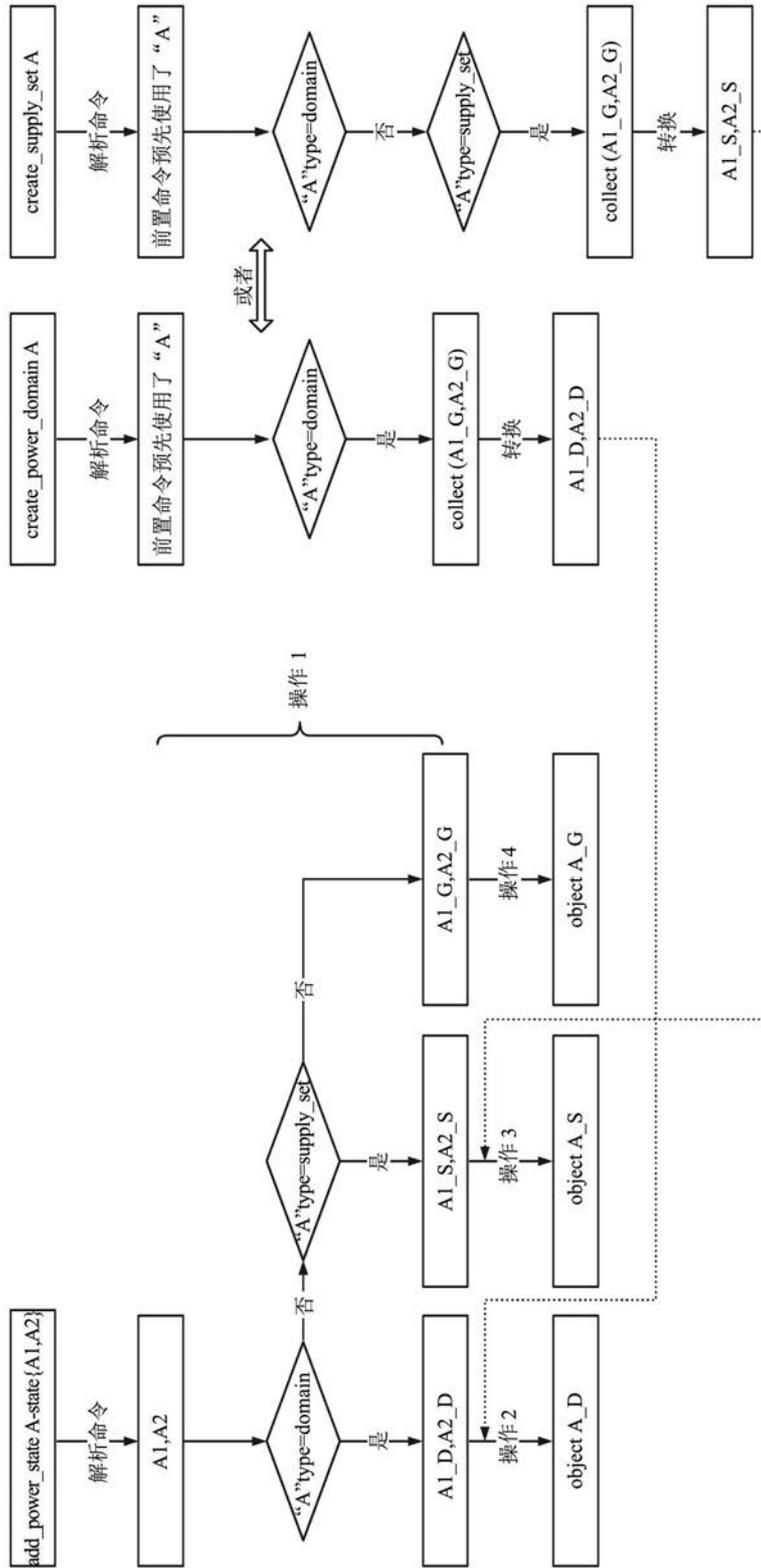


图5

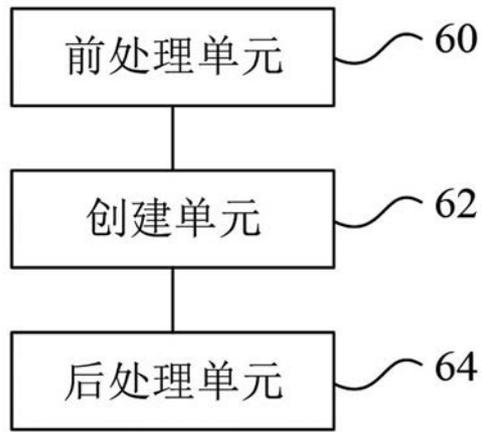


图6

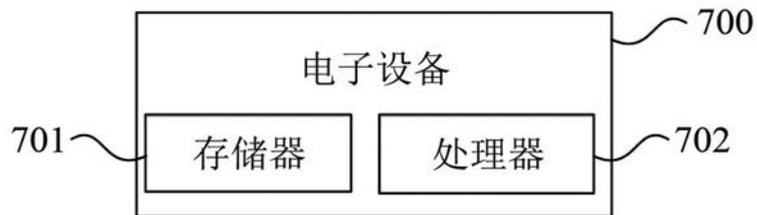


图7