(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112732636 B
(45)授权公告日 2023.05.30

(21)申请号 202110030038.3

(22)申请日 2021.01.11

(65)同一申请的已公布的文献号
申请公布号 CN 112732636 A

(43)申请公布日 2021.04.30

(73)专利权人 北京东土军悦科技有限公司
地址 100041 北京市石景山区实兴东街18号院1号楼2层01
专利权人 上海金卓科技有限公司

(72)发明人 荣超群

(74)专利代理机构 北京品源专利代理有限公司 11332
专利代理师 孟金喆

(51)Int.Cl.
*G06F 15/78*(2006.01)
*G06F 9/445*(2018.01)
*G06F 8/65*(2018.01)
*G06F 8/72*(2018.01)

(56)对比文件
CN 107293330 A,2017.10.24
CN 109491854 A,2019.03.19
CN 111125975 A,2020.05.08
CN 111651951 A,2020.09.11
CN 112100952 A,2020.12.18
CN 112183002 A,2021.01.05
JP 2011070343 A,2011.04.07
US 2006041803 A1,2006.02.23
US 6829756 B1,2004.12.07

审查员 何永海

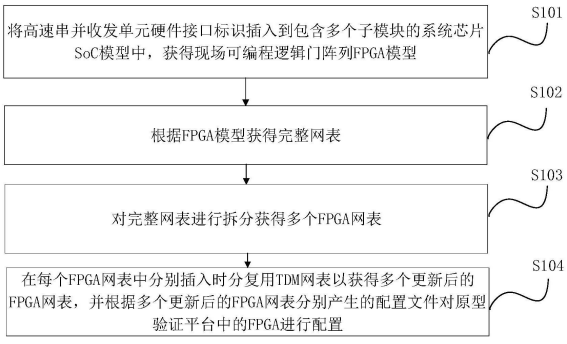
权利要求书2页 说明书10页 附图4页

(54)发明名称

# 基于多FPGA的芯片原型验证系统的配置方法、装置和设备

(57)摘要

本发明实施例公开了一种基于多FPGA的芯片原型验证系统的配置方法、装置和设备，方法包括：将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型；根据FPGA模型获得完整网表；对完整网表进行拆分获得多个FPGA网表；在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对FPGA进行配置。通过在网表阶段将包含IO接口标识的TDM以网表的形式进行插入，而不需要复杂的FPGARTL代码工作，并且将SoC模型中的总线采用高速串并收发单元硬件接口进行传输，减轻了IO接口的传输压力，从而维持了原型验证平台的时钟频率，提高了SoC芯片原型验证平台的构建效率。




CN 112732636 B

1.一种基于多FPGA的芯片原型验证平台的配置方法,其特征在于,包括:

将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;

根据所述FPGA模型获得完整网表,其中,所述完整网表中包含所述高速串并收发单元硬件接口标识与总线的对应关系;

对所述完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应所述SoC模型中的至少一个子模块;

在每个所述FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对所述原型验证平台中的FPGA进行配置,其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

2.根据权利要求1所述的方法,其特征在于,所述将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,包括:

从原型验证平台中选取高速串并收发单元硬件接口标识;

在所述SoC模型的总线界面插入所述高速串并收发单元硬件接口标识,以获得所述FPGA模型。

3.根据权利要求1所述的方法,其特征在于,所述根据所述FPGA模型获得完整网表,包括:

采用电子设计自动化EDA工具对所述FPGA模型进行参数识别;

根据参数识别结果对所述FPGA模型进行转换,获得所述完整网表。

4.根据权利要求1所述的方法,其特征在于,所述根据多个更新后的FPGA网表分别产生的配置文件对所述原型验证平台中的FPGA进行配置,包括:

根据每个更新后的FPGA网表分别获得原型验证平台中每个FPGA上的配置文件,其中,所述配置文件中包含原型验证平台中每个FPGA的高速串并收发单元硬件接口标识与总线的对应关系,以及原型验证平台中每个FPGA的IO接口标识与非总线的对应关系;

采用所述配置文件对所述原型验证平台中的每个FPGA分别进行配置。

5.根据权利要求4所述的方法,其特征在于,所述采用所述配置文件对所述原型验证平台中的每个FPGA分别进行配置,包括:

针对每个配置文件将子模块间的总线采用所对应的FPGA的高速串并收发单元硬件接口进行互联;

针对每个配置文件将子模块间的非总线采用所对应的FPGA的IO接口进行互联。

6.根据权利要求5所述的方法,其特征在于,所述高速串并收发单元硬件接口的数据传输速率大于所述IO接口的数据传输速率。

7.根据权利要求1所述的方法,其特征在于,所述高速串并收发单元硬件接口标识包括:高速串并收发单元硬件接口IP。

8.一种基于多FPGA的芯片原型验证系统的配置装置,其特征在于,包括:

FPGA模型获取模块,用于将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;

完整网表获取模块，用于根据所述FPGA模型获得完整网表，其中，所述完整网表中包含所述串并收发单元硬件接口标识与总线的对应关系；

完整网表拆分模块，用于对所述完整网表进行拆分获得多个FPGA网表，其中，每个FPGA网表分别对应所述SoC模型中的至少一个子模块；

原型验证平台配置模块，用于在每个所述FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对所述原型验证平台中的FPGA进行配置，其中，每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

9.一种电子设备，其特征在于，所述电子设备包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，

当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如权利要求1-7中任一所述的方法。

10.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，该程序被处理器执行时实现如权利要求1-7中任一所述的方法。

# 基于多FPGA的芯片原型验证系统的配置方法、装置和设备

## 技术领域

[0001] 本发明实施例涉及芯片技术领域，尤其涉及一种基于多FPGA的芯片原型验证系统的配置方法、装置和设备。

## 背景技术

[0002] 目前在芯片，例如系统芯片(System On Chip，SoC)设计和验证过程中通常需要使用现场可编程逻辑门阵列(Field Programmable Gate Array，FPGA)原型验证平台。但是当待原型验证的SoC规模超出单个FPGA具有的逻辑资源限制时，通常采用的第一种方式是对SoC进行裁剪，即将SoC中的部分子模块进行裁剪、移除进而减小待原型验证SoC的规模，并采用多个单FPGA平台实现对SoC中所有子模块的原型验证覆盖。但是这种对SoC进行裁剪的方式，无法将完整SoC中的子模块适配到统一的FPGA原型验证平台，需要多套原型验证平台配合，从而造成原型验证效率低甚至无法实现系统级的验证覆盖。

[0003] 针对上述方式所存在的问题，提出了第二种分割方式，即采用多FPGA原型验证平台，预估SoC中各个子模块的规模，在设计FPGA RTL代码阶段就将各子模块分别划分到多FPGA中，将一个完整的SoC按照子模块的规模进行划分进而适配到多FPGA中。由于SoC中各子模块之间会有大量的逻辑连接线，但FPGA的互联IO管脚数量有限，因此分割过程中通常需要对这些连接线采用时分复用(Time Division Multiplexing，TDM)的方式进行压缩。虽然第二种方式可以将完整SoC中的子模块适配到一个统一的多FPGA原型验证平台中，但IO管脚数量和TDM的使用限制了多FPGA原型验证平台的时钟频率，并且在多FPGA原型验证平台配置过程中存在大量的非芯片设计所需要的FPGARTL代码工作，从而降低了SoC芯片原型验证平台的构建效率。

## 发明内容

[0004] 本发明实施例提供了一种基于多FPGA的芯片原型验证平台的配置方法、装置和设备，以实现提高SoC芯片原型验证平台的构建效率。

[0005] 第一方面，本发明实施例提供了一种基于多FPGA的芯片原型验证平台的配置方法，包括：

[0006] 将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型，其中，子模块之间采用总线和非总线进行逻辑连接；

[0007] 根据FPGA模型获得完整网表，其中，完整网表中包含串并收发单元硬件接口标识与总线的对应关系；

[0008] 对完整网表进行拆分获得多个FPGA网表，其中，每个FPGA网表分别对应SoC模型中的至少一个子模块；

[0009] 在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置，其

中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0010] 第二方面,本发明实施例提供了一种基于多FPGA的芯片原型验证系统的配置装置,包括:

[0011] FPGA模型获取模块,用于将串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;

[0012] 完整网表获取模块,用于根据FPGA模型获得完整网表,其中,完整网表中包含串并收发单元硬件接口标识与总线的对应关系;

[0013] 完整网表拆分模块,用于对完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应SoC模型中的至少一个子模块;

[0014] 原型验证平台配置模块,用于在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置,其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0015] 第三方面,本发明实施例提供了一种电子设备,电子设备包括:

[0016] 一个或多个处理器;

[0017] 存储装置,用于存储一个或多个程序,

[0018] 当一个或多个程序被一个或多个处理器执行,使得一个或多个处理器实现如上所述的方法。

[0019] 第四方面,本发明实施例提供了一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如上所述的方法。

[0020] 本发明实施例的技术方案,通过在网表阶段将包含IO接口标识的TDM以网表的形式进行插入,而不需要复杂的FPGA RTL代码工作,并且将SoC模型中的总线采用高速串并收发单元硬件接口进行传输,减轻了IO接口的传输压力,从而维持了原型验证平台的时钟频率,提高了SoC芯片原型验证平台的构建效率。

## 附图说明

[0021] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,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相关的附图。

[0022] 图1A是本发明实施例一提供的基于多FPGA的芯片原型验证平台的配置方法的流程图;

[0023] 图1B是本发明实施例一提供的SoC模型的架构示意图;

[0024] 图1C是本发明实施例一提供的基于多FPGA的芯片原型验证平台的配置逻辑示意图;

[0025] 图1D是本发明实施例一提供的配置完成的基于多FPGA的芯片原型验证平台的架构示意图;

[0026] 图2是本发明实施例二提供的基于多FPGA的芯片原型验证平台的配置方法的流程图；

[0027] 图3是本发明实施例三提供的基于多FPGA的芯片原型验证平台的配置装置的结构示意图；

[0028] 图4是本发明实施例四提供的一种电子设备的结构示意图。

## 具体实施方式

[0029] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅用于解释本发明，而非对本发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本发明相关的部分而非全部结构。

[0030] 在更加详细地讨论示例性实施例之前应当提到的是，一些示例性实施例被描述成作为流程图描绘的处理或方法。虽然流程图将各项操作(或步骤)描述成顺序的处理，但是其中的许多操作可以被并行地、并发地或者同时实施。此外，各项操作的顺序可以被重新安排。当其操作完成时所述处理可以被终止，但是还可以具有未包括在附图中的附加步骤。所述处理可以对应于方法、软件实现、硬件实现等等。

[0031] 实施例一

[0032] 图1A是本发明实施例提供的基于多FPGA的芯片原型验证平台的配置方法的流程图，本实施例可适用于对基于多FPGA的芯片原型验证平台进行调试的情况，该方法可以由本发明实施例中的基于多FPGA的芯片原型验证平台的配置装置来执行，该装置可以采用软件和/或硬件的方式实现。如图1A所示，该方法具体包括如下操作：

[0033] 步骤S101，将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型。

[0034] 其中，SoC模型和FPGA模型都以寄存器转移级(Register-Transfer-Level，RTL)代码的形式表示，并且子模块之间采用总线和非总线进行逻辑连接。如图1B所示为SoC模型的架构示意图，其中图1B中以SoC模型中包含：通信子模块0、通信子模块1和SoC其它组成三个子模块为例进行的说明。并且相连子模块之间采用总线和非总线进行逻辑连接，例如，图1B中通信子模块0和SoC其他组件之间包括三组总线：AXI0、AXI1和AXI2，并且每一组总线包括462根逻辑连接线，因此总共有462*3＝1386根逻辑连接线；同时还包括其传输低速信号的非总线，如中断、寄存器配置等，共包含800根逻辑连接线，并且在通信子模块1和SoC其他组成之间也是包含总线和非总线的，此处不再进行赘述。当然，本实施方式中仅是举例说明，而并不对SoC模型中所包含的子模块的数量，以及子模块之间进行逻辑连接的总线和非总线的数量进行限定。如图1C所示是与图1B的SoC模型所对应的基于多FPGA的芯片原型验证平台的配置逻辑示意图，并且在图1C中的多FPGA平台上具体是包含三个FPGA。

[0035] 可选的，将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型，可以包括：从原型验证平台中选取高速串并收发单元硬件接口标识；在SoC模型的总线界面插入高速串并收发单元硬件接口标识，以获得FPGA模型。

[0036] 可选的，高速串并收发单元硬件接口标识包括：高速串并收发单元硬件接口IP。

[0037] 具体的说，本实施方式中的多FPGA的芯片原型验证平台不仅支持输入输出IO接

口,而且也支持更多更高速的SERDES硬件接口,即高速串并收发单元硬件接口。通过从原型验证平台中选取高速串并收发单元硬件接口标识,其中,标识具体可以是高速串并收发单元硬件接口的IP,并在SoC模型的总线界面插入高速串并收发单元硬件接口IP,由于SoC模型是以RTL代码的形式进行表示,因此通过将高速串并收发单元硬件接口IP插入SoC模型所获取的FPGA模型,依然是以RTL代码的形式进行表示。

[0038] 步骤S102,根据FPGA模型获得完整网表,其中,完整网表中包含高速串并收发单元硬件接口标识与总线的对应关系。

[0039] 可选的,根据FPGA模型获得完整网表,可以包括:采用电子设计自动化EDA工具对FPGA模型进行参数识别;根据参数识别结果对FPGA模型进行转换,获得完整网表。

[0040] 具体的说,本实施方式中在获得FPGA模型之后,会采用电子设计自动化(Electronic Design Automation,EDA)工具对FPGA模型进行参数识别,并且经过综合分析之后会根据参数识别结果对FPGA模型进行转换,以获得完整的网表。并且在完整的网表中包含原型验证平台中的高速串并收发单元硬件接口标识与总线的对应的关系,即每个子模块间的总线具体可以采用哪个高速串并收发单元硬件接口进行互联。

[0041] 步骤S103,对完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应SoC模型中的至少一个子模块。

[0042] 具体的说,在获得完整网表之后,由于完整的网表中反映了SoC模型中各个参数之间的关联关系,并且位于同一子模块内参数的惯性要更大。因此通过对完整网表进行拆分可以获得多个FPGA网表,并且每个FPGA网表中分别对应SoC模型中的至少一个子模块。例如,当SoC模型中所包含的子模块的数量与原型验证平台中FPGA的数量相同时,则每个FPGA网表中可以分别对应SoC模型中的一个子模块。因此在进行完整网表的拆分时,具体可以根据用户所输入的拆分指令进行自动拆分,在拆分指令中包含当前原型验证平台中所包含的FPGA的数量,以使得根据拆分指令所获得的FPGA网表的数量与原型验证平台适配。

[0043] 步骤S104,在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置。

[0044] 其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0045] 具体的说,本实施方式中在获得每个FPGA网表之后,还会获取原型验证平台中每个FPGA所对应的TDM网表,并在每个FPGA网表中分别插入所匹配的TDM网表以获得更新后的FPGA网表,并且更新后的FPGA网表的数量是与原型验证平台所包含的FPGA数量相同。而在每个TDM网表中具体包含原型验证平台上每个FPGA的输入输出IO接口标识与非总线的对应关系。相对于传统的在FPGA RTL代码阶段插入TDM RTL,从代码的角度获取IO接口与非总线的对应关系来说,本实施方式通过在网表阶段通过定制的拆分脚本再插入TDM网表,从而实现了原本在FPGA RTL阶段需要的代码工作,转换为网表阶段的脚本自动完整,实现了平台配置效率的显著提高。

[0046] 可选的,根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置,包括:根据每个更新后的FPGA网表分别获得原型验证平台中每个FPGA上的配置文件,其中,配置文件中包含原型验证平台中每个FPGA的高速串并收发单元硬件接口

标识与总线的对应关系,以及原型验证平台中每个FPGA的IO接口标识与非总线的对应关系;采用配置文件对原型验证平台中的每个FPGA分别进行配置。

[0047] 具体的说,本实施方式中是根据每个更新后的FPGA网表分别获得原型验证平台中每个FPGA上的配置文件。在配置文件中包含原型验证平台中每个FPGA的高速串并收发单元硬件接口标识与总线的对应关系,以及原型验证平台中每个FPGA的IO接口标识与非总线的对应关系,并且每个FPGA上的配置文件具体可以是以文字描述的方式对上述内容进行记载,或者以多FPGA平台所适配的能够识别的机器语言进行记载,本实施方式中并不限定每个FPGA上的配置文件的具体展示形式,只要能够被多FPGA平台识别都是在本申请的保护范围内的。并且采用每个FPGA上的配置文件分别对原型验证平台中的每个FPGA分别进行配置,从而完成原型验证平台的构建。

[0048] 可选的,采用配置文件对原型验证平台中的每个FPGA分别进行配置,可以包括:针对每个配置文件将子模块间的总线采用所对应的FPGA的高速串并收发单元硬件接口进行互联;针对每个配置文件将子模块间的非总线采用所对应的FPGA的IO接口进行互联。

[0049] 其中,在采用配置对原型验证平台中的每个FPGA分别进行配置,具体是针对每个配置文件将子模块间的总线采用所对应的FPGA的高速串并收发单元硬件接口进行互联,同时针对每个配置文件将子模块间的非总线采用所对应的FPGA的IO接口进行互联。如图1D所示为本实施方式中的配置完成的基于多FPGA的芯片原型验证平台的架构示意图,图1B是在原型验证平台上需要进行原型验证的SoC模型,并且以原型验证平台中所包含的FPGA的数量与SoC模型中分别包含的子模块的数量相同,分别是三个。例如,以SoC模型中通信子模块0和SoC其他组件之间的总线包括AXI0、AXI1和AXI2,以及非总线TDM,并且在FPGA1配置文件中包含了原型验证平台中FPGA1的高速串并收发单元硬件接口IP-AXI2GT.0,与SoC其它组成与通信子系统0之间的总线AXI0的对应关系;同时包含了原型验证平台中FPGA1的高速串并收发单元硬件接口IP-AXI2GT.1,与SoC其它组成与通信子系统0之间的总线AXI1的对应关系;以及包含了原型验证平台中FPGA1的高速串并收发单元硬件接口IP-AXI2GT.2,与SoC其它组成与通信子系统0之间的总线AXI2的对应关系。在进行分割时主要将SoC其他组件分割到FPGA1上运行,则在对平台上的FPGA1进行配置时,将SoC其他组件与通信子系统0之间的总线AXI0采用FPGA1的高速串并收发单元硬件接口AXI2GT.0进行互联,将SoC其他组件与通信子系统0之间的总线AXI1采用FPGA1的高速串并收发单元硬件接口AXI2GT.1进行互联,将SoC其他组件与通信子系统0之间的总线AXI2采用FPGA1的高速串并收发单元硬件接口AXI2GT.2进行互联,从而实现了将子模块之间的总线采用FPGA中的高速串并收发单元硬件接口进行互联。同时将SoC其他组件与通信子系统0之间的终端配置等非总线采用FPGA1的IO接口进行互联。

[0050] 可选的,高速串并收发单元硬件接口的数据传输速率大于IO接口的数据传输速率。

[0051] 例如,原型验证平台中两两FPGA之间可用于互联的IO接口为120对低电压查分信号(Low-Voltage Differential Signaling,LVDS),由于图1B中子模块0和SoC其他组件之间包括三组总线:AXI0、AXI1和AXI2,并且每一组总线包括462根逻辑连接线,因此总共有462*3=1386根逻辑连接线;同时还包括其传输低速信号的非总线,如中断、寄存器配置等,共包含800根逻辑连接线。由于总线需要全部分配到了高速串并收发单元硬件接口上,因此

仅需要800根非总线分配到120个IO接口上，压缩比仅为800/120＝6.6。相对于现有技术中的仅采用IO接口同时进行总线和非总线的互联，所达到的压缩比为(1384+800)/120＝18.2来说，由于高速串并收发单元硬件接口的数据传输速率要远远大于IO接口，因此影响原型验证平台时钟频率的仅是非总线在IO接口上的压缩比上，因此本申请的方案能够维持原型验证平台维持较高的时钟频率。

[0052] 本发明实施例的技术方案，通过在网表阶段将包含IO接口标识的TDM以网表的形式进行插入，而不需要复杂的FPGARTL代码工作，并且将SoC模型中的总线采用高速串并收发单元硬件接口进行传输，减轻了IO接口的传输压力，从而维持了原型验证平台的时钟频率，提高了SoC芯片原型验证平台的构建效率。

[0053] 实施例二

[0054] 图2是本发明实施例提供的基于多FPGA的芯片原型验证平台的配置方法的流程图，本实施例以上述实施例为基础，在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置之后，还包括对配置完成的原型验证平台进行检测。相应的，本实施例的方法具体包括如下操作：

[0055] 步骤S201，将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型。

[0056] 可选的，将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型，可以包括：从原型验证平台中选取高速串并收发单元硬件接口标识；在SoC模型的总线界面插入高速串并收发单元硬件接口标识，以获得FPGA模型。

[0057] 步骤S202，根据FPGA模型获得完整网表，其中，完整网表中包含高速串并收发单元硬件接口标识与总线的对应关系。

[0058] 可选的，根据FPGA模型获得完整网表，可以包括：采用电子设计自动化EDA工具对FPGA模型进行参数识别；根据参数识别结果对FPGA模型进行转换，获得完整网表。

[0059] 步骤S203，对完整网表进行拆分获得多个FPGA网表，其中，每个FPGA网表分别对应SoC模型中的至少一个子模块。

[0060] 步骤S204，在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置。

[0061] 其中，每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0062] 可选的，根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置，包括：根据每个更新后的FPGA网表分别获得原型验证平台中每个FPGA上的配置文件，其中，配置文件中包含原型验证平台中每个FPGA的高速串并收发单元硬件接口标识与总线的对应关系，以及原型验证平台中每个FPGA的IO接口标识与非总线的对应关系；采用配置文件对原型验证平台中的每个FPGA分别进行配置。

[0063] 可选的，采用配置文件对原型验证平台中的每个FPGA分别进行配置，可以包括：针对每个配置文件将子模块间的总线采用所对应的FPGA的高速串并收发单元硬件接口进行

互联;针对每个配置文件将子模块间的非总线采用所对应的FPGA的IO接口进行互联。

[0064] 步骤S205,对配置完成的原型验证平台进行检测。

[0065] 其中,本实施方式中在对原型验证平台中的FPGA分别进行配置,并完成原型验证平台的构建之后,可以通过运行原型验证平台的方式进行检测,如果通过运行原型验证平台,确定无法获得验证结果,或者验证结果出现明显错误,例如,出现乱码等,此时则可能是原型验证平台配置错误或者原型验证平台硬件结构出现了故障。根据验证结果进行检测,并在确定出现异常的情况下进行报警,具体可以采用语音或文字的方式进行报警,本实施方式中并不限定报警的具体方式,以便于提醒用户及时进行设备的检修或原型验证平台的重新配置。

[0066] 本发明实施例的技术方案,通过在网表阶段将包含IO接口标识的TDM以网表的形式进行插入,而不需要复杂的FPGA RTL代码工作,并且将SoC模型中的总线采用高速串并收发单元硬件接口进行传输,减轻了IO接口的传输压力,从而维持了原型验证平台的时钟频率,提高了SoC芯片原型验证平台的构建效率。通过对配置完成的原型验证平台进行检测,并在确定检测异常的情况下及时进行报警,从而提高了原型验证平台的质量以及构建效率。

[0067] 实施例三

[0068] 图3为本发明实施例提供的基于多FPGA的芯片原型验证平台的配置装置的结构示意图,该装置包括:FPGA模型获取模块310、完整网表获取模块320、完整网表拆分模块330和原型验证平台配置模块340。

[0069] 其中,FPGA模型获取模块310,用于将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;

[0070] 完整网表获取模块320,用于根据FPGA模型获得完整网表,其中,完整网表中包含高速串并收发单元硬件接口标识与总线的对应关系;

[0071] 完整网表拆分模块330,用于对完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应SoC模型中的至少一个子模块;

[0072] 原型验证平台配置模块340,用于在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置,其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0073] 可选的,FPGA模型获取模块,用于从原型验证平台中选取高速串并收发单元硬件接口标识;

[0074] 在SoC模型的总线界面插入高速串并收发单元硬件接口标识,以获得FPGA模型。

[0075] 可选的,完整网表获取模块,用于采用电子设计自动化EDA工具对FPGA模型进行参数识别;

[0076] 根据参数识别结果对FPGA模型进行转换,获得完整网表,其中,完整网表以关联图的形式对FPGA模型进行展示。

[0077] 可选的,原型验证平台配置模块包括:配置文件获取子模块,用于根据每个更新后的FPGA网表分别获得原型验证平台中每个FPGA上的配置文件,其中,配置文件中包含原型

验证平台中每个FPGA的高速串并收发单元硬件接口标识与总线的对应关系，以及原型验证平台中每个FPGA的IO接口标识与非总线的对应关系；

[0078] 配置子模块，用于采用配置文件对原型验证平台中的每个FPGA分别进行配置。

[0079] 可选的，配置子模块，用于针对每个配置文件将子模块间的总线采用所对应的FPGA的高速串并收发单元硬件接口进行互联；

[0080] 针对每个配置文件将子模块间的非总线采用所对应的FPGA的IO接口进行互联。

[0081] 可选的，高速串并收发单元硬件接口的数据传输速率大于IO接口的数据传输速率。

[0082] 可选的，高速串并收发单元硬件接口标识包括：高速串并收发单元硬件接口IP。

[0083] 上述装置可执行本发明任意实施例所提供的基于多FPGA的芯片原型验证平台的配置方法，具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节，可参见本发明任意实施例提供的方法。

[0084] 实施例四

[0085] 图4是本发明实施例提供的一种电子设备的结构示意图。图4示出了适用于用来实现本发明实施方式的示例性电子设备412的框图。图4显示的电子设备412仅仅是一个示例，不应对本发明实施例的功能和使用范围带来任何限制。

[0086] 如图4所示，电子设备412以通用计算设备的形式出现。电子设备412的组件可以包括但不限于：一个或者多个处理器416，存储器428，连接不同系统组件(包括存储器428和处理器416)的总线418。

[0087] 总线418表示几类总线结构中的一种或多种，包括存储器总线或者存储器控制器，外围总线，图形加速端口，处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说，这些体系结构包括但不限于工业标准体系结构(ISA)总线，微通道体系结构(MAC)总线，增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0088] 电子设备412典型地包括多种计算机系统可读介质。这些介质可以是任何能够被电子设备412访问的可用介质，包括易失性和非易失性介质，可移动的和不可移动的介质。

[0089] 存储器428用于存储指令。存储器428可以包括易失性存储器形式的计算机系统可读介质，例如随机存取存储器(RAM)430和/或高速缓存存储器432。电子设备412可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例，存储系统434可以用于读写不可移动的、非易失性磁介质(图4未显示，通常称为“硬盘驱动器”)。尽管图4中未示出，可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器，以及对可移动非易失性光盘(例如CD-ROM，DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下，每个驱动器可以通过一个或者多个数据介质接口与总线418相连。存储器428可以包括至少一个程序产品，该程序产品具有一组(例如至少一个)程序模块，这些程序模块被配置以执行本发明各实施例的功能。

[0090] 具有一组(至少一个)程序模块442的程序/实用工具440，可以存储在例如存储器428中，这样的程序模块442包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块442通常执行本发明所描述的实施例中的功能和/或方法。

[0091] 电子设备412也可以与一个或多个外部设备414(例如键盘、指向设备、显示器424

等)通信,还可与一个或者多个使得用户能与该电子设备412交互的设备通信,和/或与使得该电子设备412能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口422进行。并且,电子设备412还可以通过网络适配器420与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图所示,网络适配器420通过总线418与电子设备412的其它模块通信。应当明白,尽管图4中未示出,可以结合电子设备412使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0092] 处理器416通过运行存储在存储器428中的指令,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的基于多FPGA的芯片原型验证平台的配置方法:将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;根据FPGA模型获得完整网表,其中,完整网表中包含高速串并收发单元硬件接口标识与总线的对应关系;对完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应SoC模型中的至少一个子模块;在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置,其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0093] 实施例五

[0094] 本发明实施例提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如本申请所有发明实施例提供的基于多FPGA的芯片原型验证平台的配置方法:

[0095] 将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中,获得现场可编程逻辑门阵列FPGA模型,其中,子模块之间采用总线和非总线进行逻辑连接;根据FPGA模型获得完整网表,其中,完整网表中包含高速串并收发单元硬件接口标识与总线的对应关系;对完整网表进行拆分获得多个FPGA网表,其中,每个FPGA网表分别对应SoC模型中的至少一个子模块;在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表,并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置,其中,每个TDM网表中包含原型验证平台中每个FPGA的输入输出IO接口标识与非总线的对应关系。

[0096] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、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。在本文件中,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。

[0097] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,

其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0098] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0099] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络包括局域网(LAN)或广域网(WAN)连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0100] 注意，上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解，本发明不限于这里所述的特定实施例，对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此，虽然通过以上实施例对本发明进行了较为详细的说明，但是本发明不仅仅限于以上实施例，在不脱离本发明构思的情况下，还可以包括更多其他等效实施例，而本发明的范围由所附的权利要求范围决定。

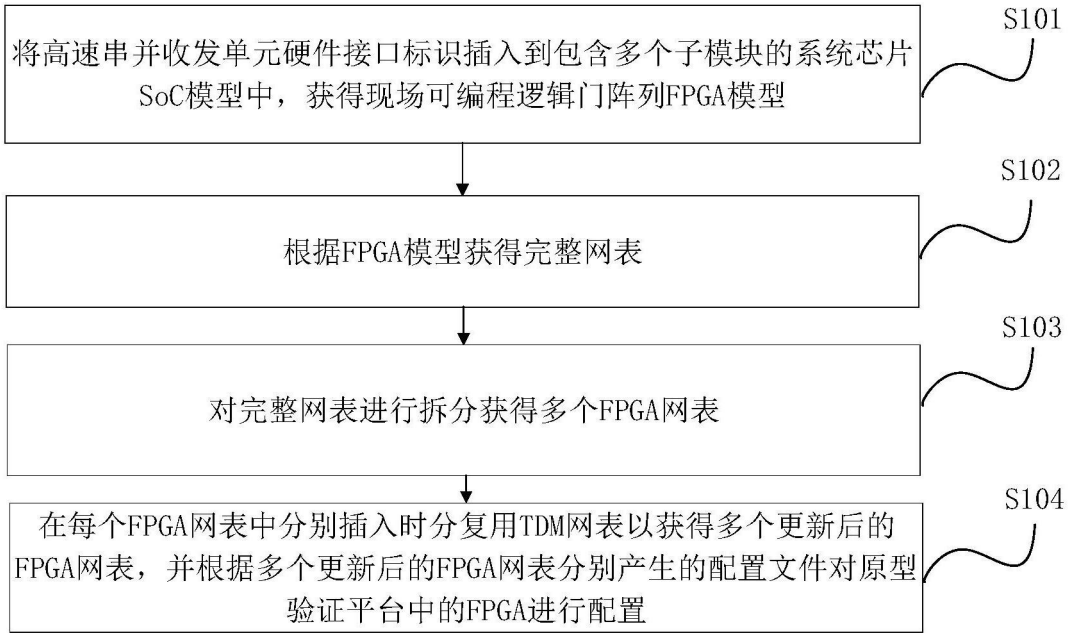

将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型
S101
根据FPGA模型获得完整网表
S102
对完整网表进行拆分获得多个FPGA网表
S103
在每个FPGA网表中分别插入时分复用TDM网表以获得多个更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置
S104

图1A

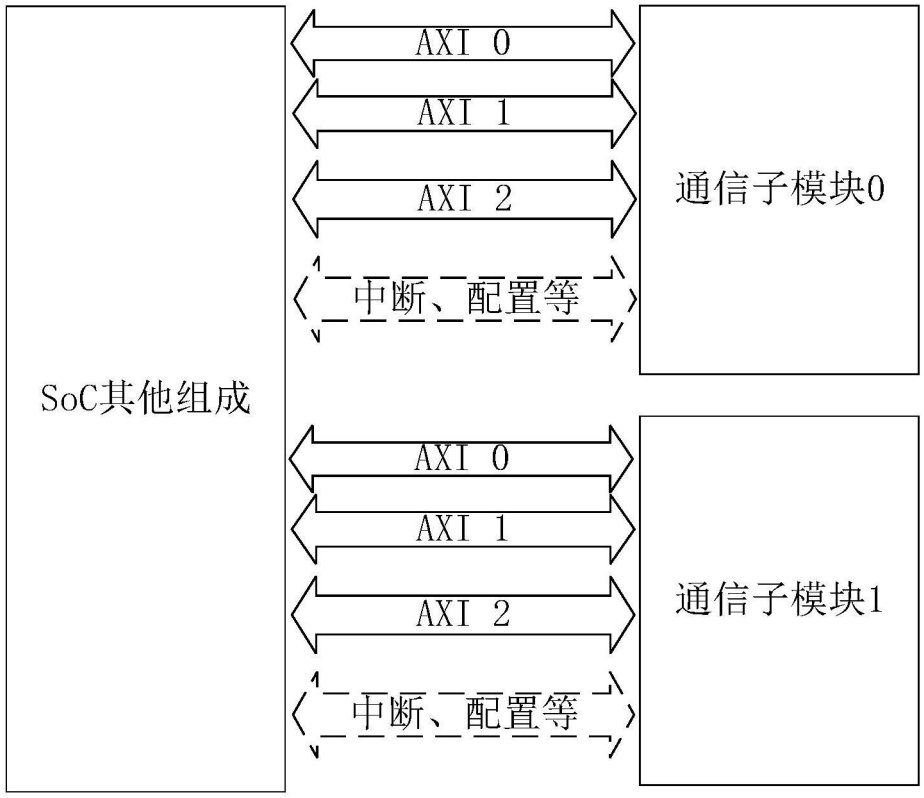

SoC其他组成
AXI 0
AXI 1
AXI 2
中断、配置等
通信子模块0
AXI 0
AXI 1
AXI 2
中断、配置等
通信子模块1

图1B

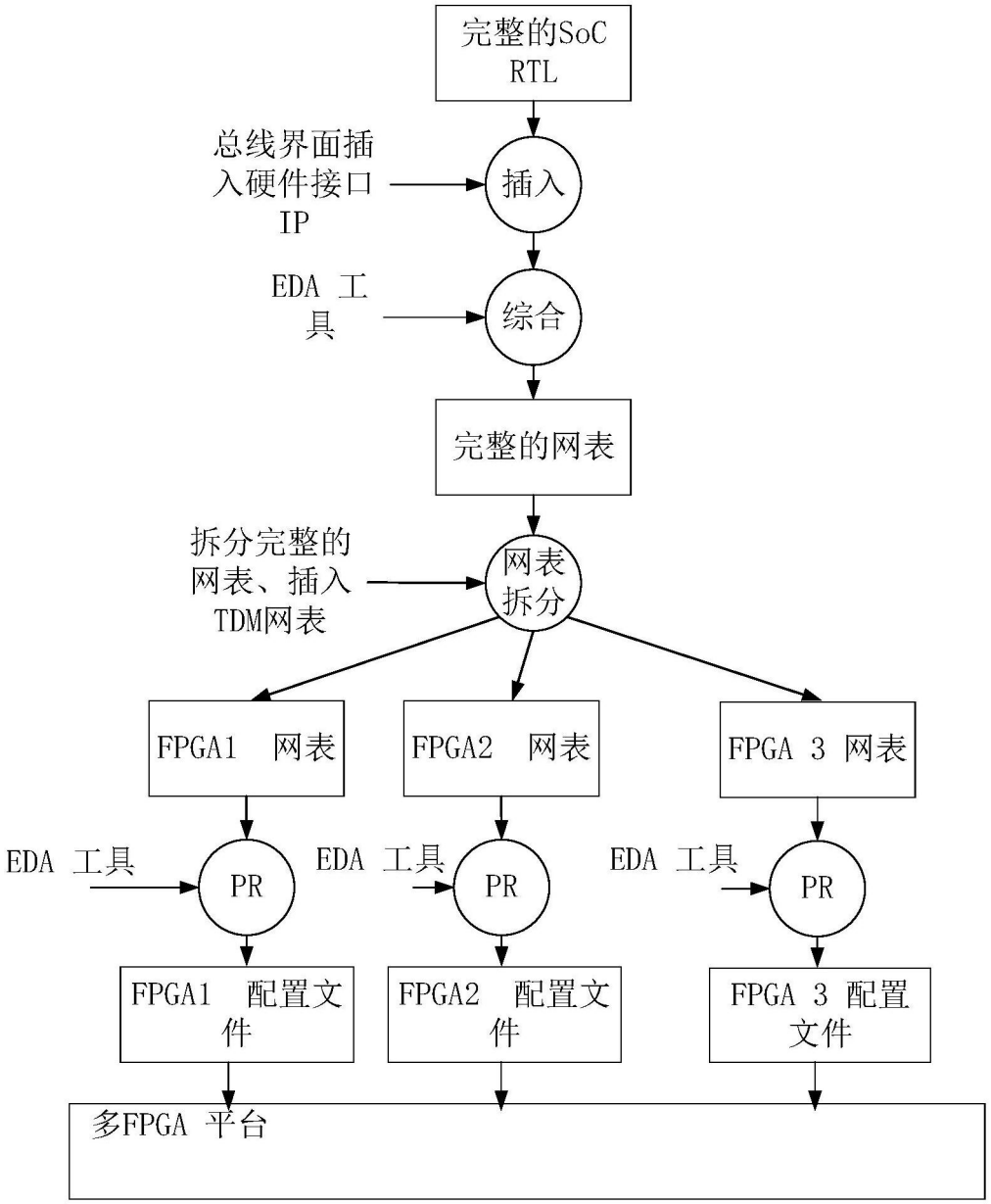

完整的SoC RTL
总线界面插入硬件接口IP
插入
EDA 工具
综合
完整的网表
拆分完整的网表、插入TDM网表
网表拆分
FPGA1 网表
FPGA2 网表
FPGA 3 网表
EDA 工具
PR
EDA 工具
PR
EDA 工具
PR
FPGA1 配置文件
FPGA2 配置文件
FPGA 3 配置文件
多FPGA 平台

图1C

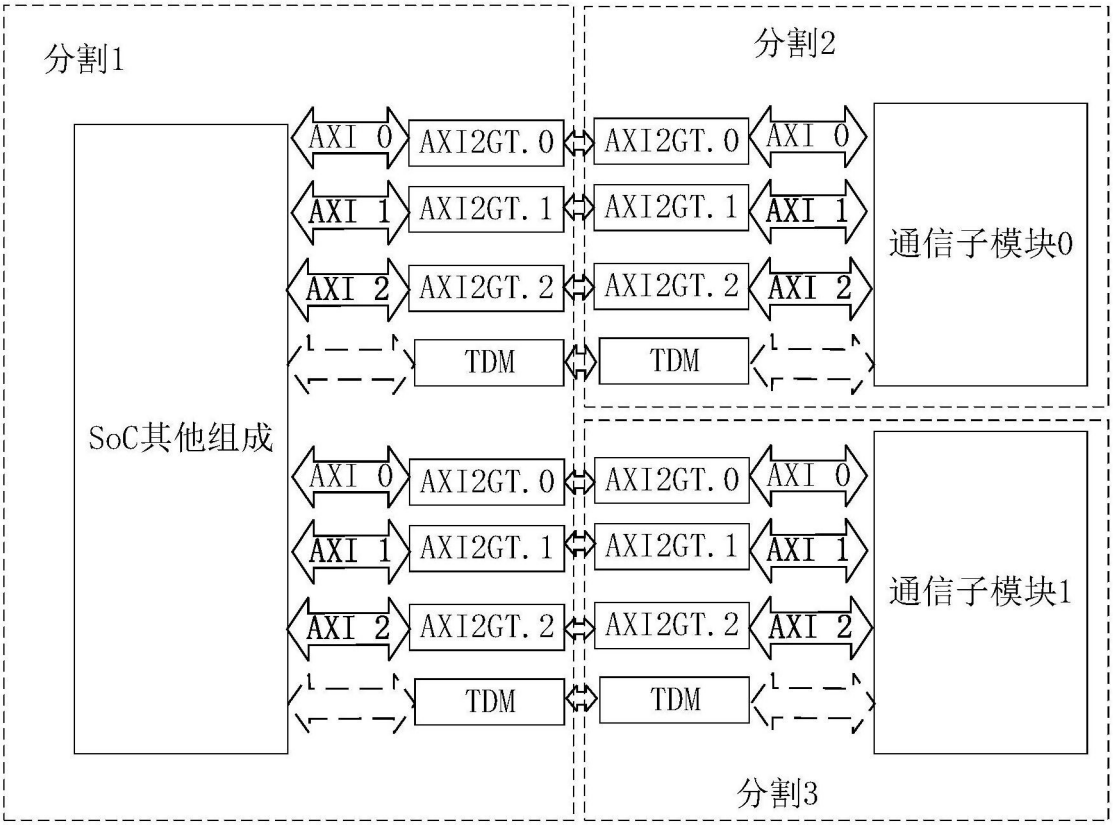

分割1
分割2
SoC其他组成
AXI 0
AXI 1
AXI 2
AXI2GT.0
AXI2GT.1
AXI2GT.2
TDM
通信子模块0
通信子模块1
分割3

图1D

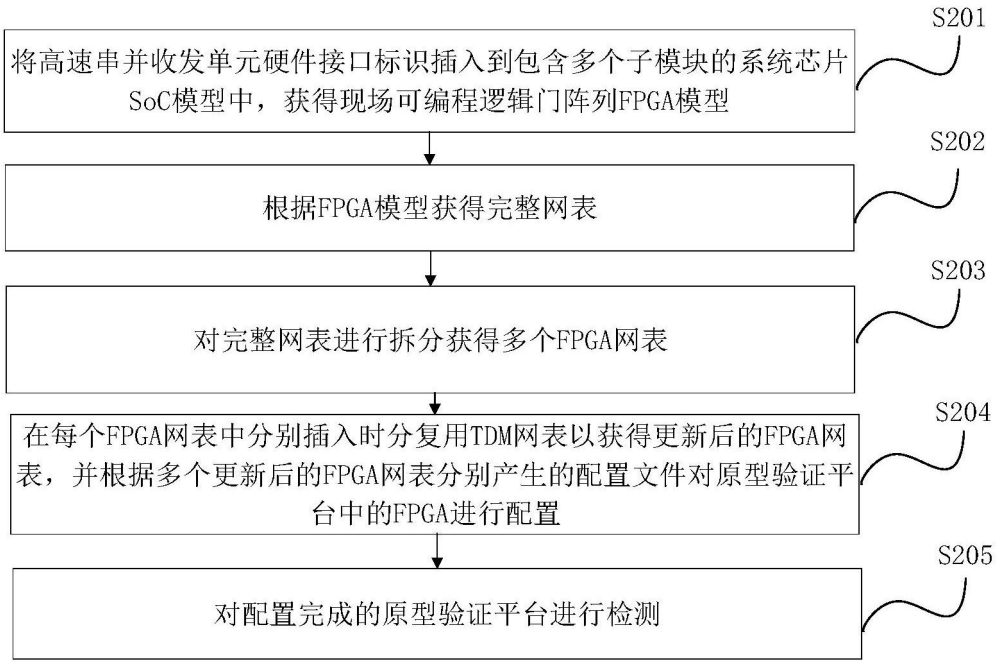

S201
将高速串并收发单元硬件接口标识插入到包含多个子模块的系统芯片SoC模型中，获得现场可编程逻辑门阵列FPGA模型
S202
根据FPGA模型获得完整网表
S203
对完整网表进行拆分获得多个FPGA网表
S204
在每个FPGA网表中分别插入时分复用TDM网表以获得更新后的FPGA网表，并根据多个更新后的FPGA网表分别产生的配置文件对原型验证平台中的FPGA进行配置
S205
对配置完成的原型验证平台进行检测

图2

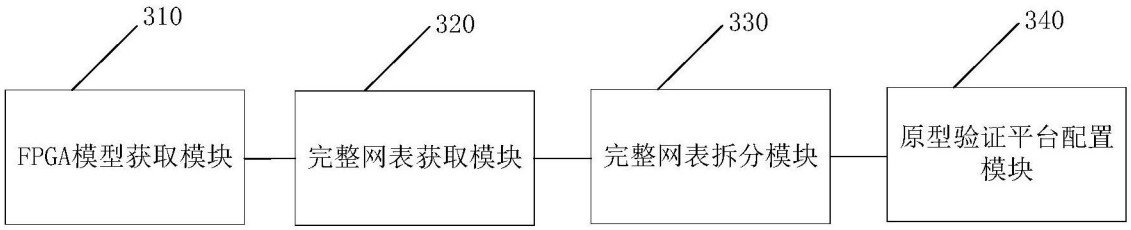

310
320
330
340
FPGA模型获取模块
完整网表获取模块
完整网表拆分模块
原型验证平台配置模块

图3

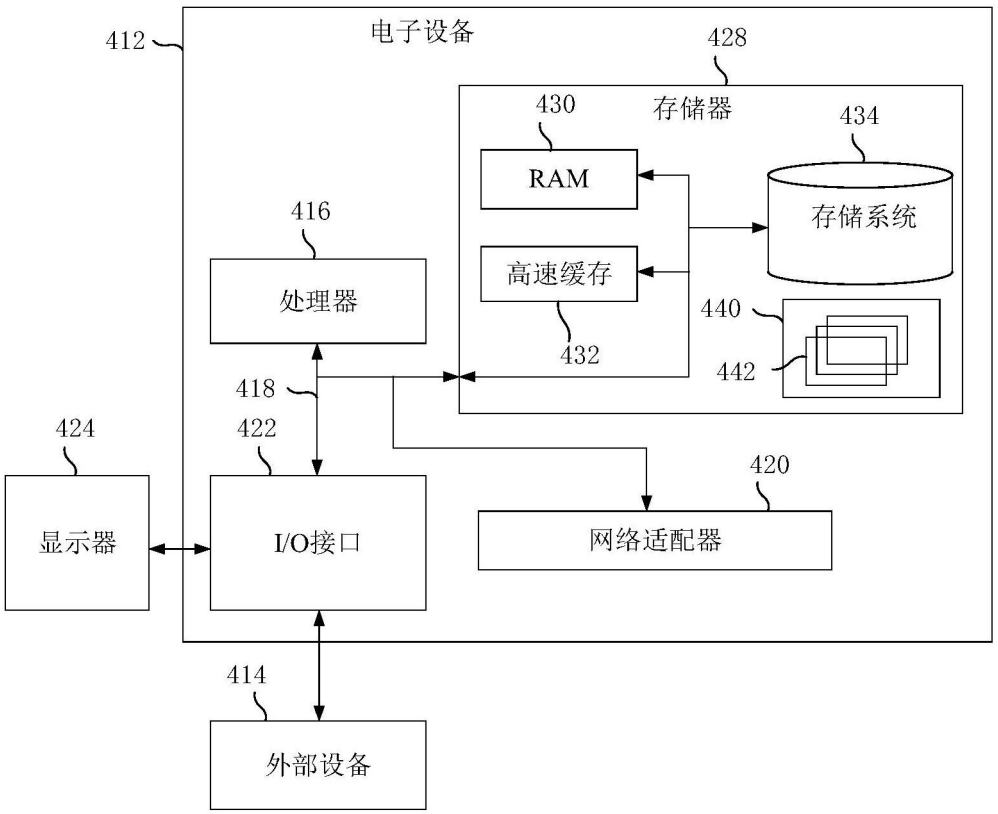

412
电子设备
428
430
存储器
434
RAM
存储系统
416
处理器
高速缓存
440
432
442
418
424
422
420
显示器
I/O接口
网络适配器
414
外部设备

图4