



(21) 申请号 202310988910.4

(22) 申请日 2023.08.07

(65) 同一申请的已公布的文献号
申请公布号 CN 117033112 A

(43) 申请公布日 2023.11.10

(73) 专利权人 西安微电子技术研究所
地址 710065 陕西省西安市雁塔区太白南路198号

专利权人 西北工业大学

(72) 发明人 匡乃亮 唐磊 刘曦 赵超
郑江滨 郭雁蓉 李珂

(74) 专利代理机构 西安通大专利代理有限责任公司
61200

专利代理师 姚咏华

(51) Int.Cl.

G06F 11/273 (2006.01)

G06F 11/22 (2006.01)

(56) 对比文件

CN 108319526 A, 2018.07.24

杜影等. “基于FPGA的板级BIST设计和实现策略”. 《计算机测量与控制》. 2008, 389-391.

审查员 王新忠

权利要求书2页 说明书8页 附图3页

(54) 发明名称

一种系统级DFT实现方法、系统、设备及介质

(57) 摘要

本发明公开了一种系统级DFT实现方法、系统、设备及介质,属于信息处理微系统设计领域,通过构建可测试性系统架构、搭建测试状态机和生成固化测试向量的方式,即通过系统架构分析优化、BIST测试控制逻辑的植入和测试向量实现等技术手段,从而实现信息处理微系统高度自测试、自评价,达到脱离外部测试设备的完全自测试,以满足在微系统研制过程中的测试筛选、故障分析定位和系统应用中的功能自检和软件智能化的要求;采用本方法开发的系统级DFT功能,功能覆盖率可做到100%,设计可行性高,测试设备依赖程度低,自动化程度高,代码继承性强。不改变原有线路拓扑和负载,不增加封装内信号布线复杂度,不增加产品工艺复杂度和难度,普适性高、实现成本低。

构建以FPGA为核心的可测试性系统架构

根据可测试性系统架构搭建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机

利用测试状态机完成FPGA以及外围芯片的自检

1. 一种系统级DFT实现方法,其特征在于,包括:
构建以FPGA为核心的可测试性系统架构;
根据可测试性系统架构搭建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;
利用测试状态机完成FPGA以及外围芯片的自检;
其中,所述外围芯片采用由处理器、处理器的用于boot的ROM、处理器运行的RAM、FPGA以及FPGA的用于boot的ROM所构成的信息处理微系统;
构建以FPGA为核心的可测试性系统架构的具体步骤包括:
在FPGA的IO上接入处理器的具备BOOT功能的双向数据接口,将处理器上用于传输GPIO信号的双向数据接口以及用于BOOT启动的关键控制信号接入FPGA;
还包括:将内部电源、温度信号接入FPGA模拟输入IO;
所述测试状态机的自检具体步骤如下:
S1:进行关键模拟量自检,若合格,继续下一步,否则,自检结束;
S2:加载测试向量进行FPGA自检,存储FPGA检测结果;
S3:加载测试向量进行外围芯片自检,存储外围芯片检测结果;
S2执行前,根据外部IO状态判断FPGA自检需求,若有需求,则继续S2,否则,跳转至S3;
S3执行前,根据外部IO状态判断外围芯片自检需求,若有需求,则继续S3,否则,自检结束;
采用测试向量对FPGA以及外围芯片进行自检;其中,测试向量包括FPGA自检测试向量和处理器自检测试向量;
FPGA自检的步骤包括:
S201:修改FPGA配置寄存器中下次启动地址为FPGA自检测试向量存储首地址;
S202:启动IPI命令,使FPGA完成自复位,进入重新加载流程,加载FPGA自检测试向量并执行,完成后将FPGA自检结果存储至ROM中;
S203:判断是否完成全部FPGA自检测试向量加载,若完成,则继续S3;否则,重新执行S201;
外围芯片自检的步骤包括:
S301:修改FPGA配置寄存器中下次启动地址为处理器自检测试向量的存储首地址;
S302:再启动IPI命令,使FPGA完成自复位,进入重新加载流程,加载处理器自检测试向量并执行,在自检测试向量执行中,FPGA控制单个或多个处理器进入BOOT启动模式,并将存储在FPGA例化ROM中的数据通过BOOT通讯口加载到处理器中,完成后处理器开始运行,实现处理器各功能模块和处理器外围存储器功能的自检测试,完成测试后由FPGA收集外围芯片自检结果并将其存储至ROM中;
S303:判断是否完成全部外围处理器测试向量加载,若完成,则结束;否则,重新执行S301;
所述FPGA自检测试向量基于FPGA单芯片测试向量生成并采用MultiBoot数据流固化方式进行固化;
所述处理器自检测试向量采用最小化原则生成多份,通过将对应的.bin文件以ROM初始数据的方式加载至MCS文件中,采用MultiBoot数据流固化方式进行固化;或通过FPGA的桥接逻辑进行配置ROM直接访问固化。

2. 一种系统级DFT设计系统,用于实现权利要求1所述系统级DFT实现方法的步骤,其特征在于,包括:

构建模块,用于构建以FPGA为核心的可测试性系统架构;

自检搭建模块,用于根据可测试性系统架构构建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;

自检模块,用于利用测试状态机完成FPGA以及外围芯片的自检。

3. 一种设备,其特征在于,包括:

存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序时实现权利要求1所述系统级DFT实现方法的步骤。

4. 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时用于实现权利要求1所述系统级DFT实现方法的步骤。

一种系统级DFT实现方法、系统、设备及介质

技术领域

[0001] 本发明属于信息处理微系统设计领域,具体涉及一种系统级DFT实现方法、系统、设备及介质。

背景技术

[0002] 嵌入式计算机的小型化、模块化和及多特性电路和多功能的高密度集成,是当前系统嵌入式环境的特定需求。系统集成出现了SOC (System-On-Chip)、MCM (Multi-Chip-Module)、SiP (System-in-Package) 等技术。微系统是新一代系统集成技术,是在“松弛”约束下的全系统级集成。信息处理微系统是面向海量数据运算、大数据存储等功能性能需求的一类微型系统。其基本组装单元可以是硅芯片、铝碳化硅芯片、砷化镓芯片、硅光芯片等,甚至是微模组单元。该技术将数字与模拟信号处理电路、存储器以及光器件及无源网络等多个功能模块,高效地融合在一个封装体内,属于真正的系统级封装。它容易实现复杂系统的小型化、轻量化、高性能和高可靠性,特别适合于航天、航空、便携式电子系统等对体积、重量和环境要求苛刻的场合,成为系统集成的首选技术。

[0003] DFT (Design For Testability) 可测试性设计是在产品设计实现的过程中,充分考虑到测试的要求,通过增加一定的开销和代价,获得最大可测试性的设计过程,是为了达到故障检测目的所做的辅助性、必要性的设计。基于DFT的测试技术是提升产品测试覆盖率、增强产品质量、降低测试成本的重要手段之一。

[0004] 目前,DFT技术在以CPU、SoC、DSP等为代表的超大规模单片集成电路中应用广泛,技术成熟度高。当前DFT所采用的测试结构主要有扫描链Scan Chain、边界扫描Boundary Scan、内建自测试BIST等。Scan Chain主要针对带有记忆元件的时序电路,用于包括寄存器、组合逻辑、触发器等;Boundary Scan主要针对IO引脚,以JTAG结构应用最为广泛,通过在芯片的IO管脚和核心电路之间增加边界扫描单元,将它们串联成扫描链,在测试控制器的控制下,实现IO管脚信号状态设定和读取,从而验证芯片IO功能,以及扩展多芯片之间IO互连性测试;BIST是一种可测试性设计方法,它的基本思想是由电路自动生成测试向量,不需要外界施加测试向量,然后通过自身逻辑判断所得的测试响应是否正确,极大的降低了对测试设备的要求,该项技术主要应用于存储器中,形成一个重要分支MBIST。

[0005] 微系统测试技术是微系统研制过程中不可或缺的重要环节,是保障微系统产品质量的关键因素。传统的二维集成电路(2D IC)测试技术只适用于单层芯片,基于多芯片微系统的测试技术还不成熟,大部分测试问题仍处于未开发状态。在微系统可测试性设计(DFT)方面,目前主流技术是增加边界扫描链数量,增大扫描的IO覆盖率,从功能测试上,主要采用系统级测试方法,即所有可编程器件固化唯一的综合性测试程序,通过上位机的通讯控制实现各个独立功能以及多个功能耦合的功能性能测试。

发明内容

[0006] 为克服上述技术的缺点,本发明提供一种系统级DFT实现方法、系统、设备及介质,

能够解决现有测试技术必须借助外部测试设备才能完成信息处理微系统的测试和评价的技术问题。

[0007] 为了达到上述目的,本发明采用技术方案如下:

[0008] 一种系统级DFT实现方法,包括:

[0009] 构建以FPGA为核心的可测试性系统架构;

[0010] 根据可测试性系统架构搭建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;

[0011] 利用测试状态机完成FPGA以及外围芯片的自检。

[0012] 进一步地,构建以FPGA为核心的可测试性系统架构的具体步骤包括:

[0013] 在FPGA的IO上接入处理器的具备BOOT功能的双向数据接口,将处理器上用于传输GPIO信号的双向数据接口以及用于BOOT启动的关键控制信号接入FPGA。

[0014] 进一步地,还包括;将内部电源、温度信号接入FPGA模拟输入IO。

[0015] 进一步地,所述测试状态机的自检具体步骤如下:

[0016] S1:进行关键模拟量自检,若合格,继续下一步,否则,自检结束;

[0017] S2:加载测试向量进行FPGA自检,存储FPGA检测结果;

[0018] S3:加载测试向量进行外围芯片自检,存储外围芯片检测结果。

[0019] 进一步地,

[0020] S2执行前,根据外部IO状态判断FPGA自检需求,若有需求,则继续S2,否则,跳转至S3;

[0021] S3执行前,根据外部IO状态判断外围芯片自检需求,若有需求,则继续S3,否则,自检结束。

[0022] 进一步地,采用测试向量对FPGA以及外围芯片进行自检;其中,测试向量包括FPGA自检测试向量和处理器自检测试向量;

[0023] FPGA自检的步骤包括:

[0024] S201:修改FPGA配置寄存器中下次启动地址为FPGA自检测试向量存储首地址;

[0025] S202:启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载FPGA自检测试向量并执行,完成后将FPGA自检结果存储至ROM中;

[0026] S203:判断是否完成全部FPGA自检测试向量加载,若完成,则继续S3;否则,重新执行S201;

[0027] 外围芯片自检的步骤包括:

[0028] S301:修改FPGA配置寄存器中下次启动地址为处理器自检测试向量的存储首地址;

[0029] S302:再启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载处理器自检测试向量并执行,在自检测测试向量执行中,FPGA控制单个或多个处理器进入BOOT启动模式,并将存储在FPGA例化ROM中的数据通过BOOT通讯口加载到处理器中,完成后处理器开始运行,实现处理器各功能模块和处理器外围存储器功能的自检测测试,完成测试后由FPGA收集外围芯片自检结果并将其存储至ROM中;

[0030] S303:判断是否完成全部外围处理器测试向量加载,若完成,则结束;否则,重新执行S301。

- [0031] 进一步地,
- [0032] 所述FPGA自检测测试向量基于FPGA单芯片测试向量生成并采用MultiBoot数据流固化方式进行固化;
- [0033] 所述处理器自检测测试向量采用最小化原则生成多份,通过将对应的.bin文件以ROM初始数据的方式加载至MCS文件中,采用MultiBoot数据流固化方式进行固化;或通过FPGA的桥接逻辑进行配置ROM直接访问固化。
- [0034] 一种系统级DFT设计系统,用于实现上述系统级DFT实现方法的步骤,其特征在于,包括:
- [0035] 构建模块,用于构建以FPGA为核心的可测试性系统架构;
- [0036] 自检搭建模块,用于根据可测试性系统架构构建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;
- [0037] 自检模块,用于利用测试状态机完成FPGA以及外围芯片的自检。
- [0038] 一种设备,包括:
- [0039] 存储器,用于存储计算机程序;
- [0040] 处理器,用于执行所述计算机程序时实现上述系统级DFT实现方法的步骤。
- [0041] 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时用于实现上述系统级DFT实现方法的步骤。
- [0042] 相比于现有技术,本发明具有有益效果如下:
- [0043] 本发明提供一种系统级DFT实现方法,本发明通过构建可测试性系统架构、搭建测试状态机和生成固化测试向量的方式,实现了系统级DFT,即通过系统架构分析优化、BIST测试控制逻辑的植入和测试向量实现等技术手段,从而实现信息处理微系统高度自测试、自评价,达到脱离外部测试设备的完全自测试,以满足在微系统研制过程中的测试筛选、故障分析定位和系统应用中的功能自检和软件智能化的要求;采用本方法开发的系统级DFT功能,功能覆盖率可做到100%,设计可行性高,测试设备依赖程度低,自动化程度高,代码继承性强。不改变原有线路拓扑和负载,不增加封装内信号布线复杂度,不增加产品工艺复杂度和难度,普适性高、实现成本低。

附图说明

- [0044] 图1为本发明实施例提供的信息处理系统微系统的架构结构示意图;
- [0045] 图2为本发明实施例提供的信息处理系统微系统的产品结构示意图;
- [0046] 图3为本发明实施例提供的测试状态机的自检的流程示意图;
- [0047] 图4为本发明提供的一种系统级DFT实现方法的流程图;
- [0048] 图5为本发明提供的一种系统级DFT设计系统的结构示意图。
- [0049] 附图标记为:
- [0050] 封装基板-100;热沉-101;第一TSV中介层-104;第二TSV中介层-107;焊球-108;微凸点-109;FPGA-200;第一信号-209;处理器-210;第一存储器-212;第二存储器-222;第二信号-219;第三信号-230。

具体实施方式

[0051] 本发明提供一种系统级DFT实现方法,如图4所示,包括如下步骤:

[0052] 一种系统级DFT实现方法,包括:构建以FPGA为核心的可测试性系统架构;根据可测试性系统架构搭建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;利用测试状态机完成FPGA以及外围芯片的自检。

[0053] 其中,构建以FPGA为核心的可测试性系统架构的具体步骤包括:

[0054] 在FPGA的IO上接入处理器的具备BOOT功能的双向数据接口,将处理器上用于传输GPIO信号的双向数据接口以及用于BOOT启动的关键控制信号接入FPGA。将内部电源、温度信号接入FPGA模拟输入IO。

[0055] 上述测试状态机的自检具体步骤如下:

[0056] S1:进行关键模拟量自检,若合格,继续下一步,否则,自检结束;其中,关键模拟量包括微系统电源模拟量和微系统温度模拟量。

[0057] S2:加载测试向量进行FPGA自检,存储FPGA检测结果;

[0058] S3:加载测试向量进行外围芯片自检,存储外围芯片检测结果。

[0059] 特别的,在S2执行前,根据外部IO状态判断FPGA自检需求,若有需求,则继续S2,否则,跳转至S3;

[0060] 在S3执行前,根据外部IO状态判断外围芯片自检需求,若有需求,则继续S3,否则,自检结束。

[0061] 采用测试向量对FPGA以及外围芯片进行自检,上述提及的测试向量包括FPGA自检测试向量和处理器自检测试向量;

[0062] 具体的,FPGA自检的步骤包括:

[0063] S201:修改FPGA配置寄存器中下次启动地址为FPGA自检测试向量存储首地址;

[0064] S202:启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载FPGA自检测试向量并执行,完成后将FPGA自检结果存储至ROM中;

[0065] S203:判断是否完成全部FPGA自检测试向量加载,若完成,则继续S3;否则,重新执行S201;

[0066] 外围芯片自检的步骤包括:

[0067] S301:修改FPGA配置寄存器中下次启动地址为处理器自检测试向量的存储首地址;

[0068] S302:再启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载处理器自检测试向量并执行,在自检测试向量执行中,FPGA控制单个或多个处理器进入BOOT启动模式,并将存储在FPGA例化ROM中的数据通过BOOT通讯口加载到处理器中,完成后处理器开始运行,实现处理器各功能模块和处理器外围存储器功能的自检测试,完成测试后由FPGA收集外围芯片自检结果并将其存储至ROM中;

[0069] S303:判断是否完成全部外围处理器测试向量加载,若完成,则结束;否则,重新执行S301。

[0070] 所述FPGA自检测试向量基于FPGA单芯片测试向量生成并采用MultiBoot数据流固化方式进行固化;

[0071] 所述处理器自检测试向量采用最小化原则生成多份,通过将对应的.bin文件以

ROM初始数据的方式加载至MCS文件中,采用MultiBoot数据流固化方式进行固化;或通过FPGA的桥接逻辑进行配置ROM直接访问固化。

[0072] 本发明还提供了一种设备,包括:存储器,用于存储计算机程序;处理器,用于执行所述计算机程序时实现所述的系统级DFT实现方法的步骤。

[0073] 所述处理器执行所述计算机程序时实现上述系统级DFT设计的步骤,例如:构建以FPGA为核心的可测试性系统架构;根据可测试性系统架构构建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;利用测试状态机完成FPGA以及外围芯片的自检。

[0074] 或者,所述处理器执行所述计算机程序时实现上述系统中各模块的功能,例如构建模块,用于构建以FPGA为核心的可测试性系统架构;自检搭建模块,用于根据可测试性系统架构构建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;自检模块,用于利用测试状态机完成FPGA以及外围芯片的自检。

[0075] 示例性的,所述计算机程序可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器中,并由所述处理器执行,以完成本发明。所述一个或多个模块/单元可以是能够完成预设功能的一系列计算机程序指令段,所述指令段用于描述所述计算机程序在所述系统级DFT设计设备中的执行过程。例如,所述计算机程序可以被分割成数据转换分析模块、展示标注模块和数据关联模块;各模块具体功能如下:构建模块,用于构建以FPGA为核心的可测试性系统架构;自检搭建模块,用于根据可测试性系统架构构建FPGA内部的用于FPGA自检和外围芯片自检的测试状态机;自检模块,用于利用测试状态机完成FPGA以及外围芯片的自检。

[0076] 所述系统级DFT设计设备可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述系统级DFT设计设备可包括,但不仅限于处理器、存储器。本领域技术人员可以理解,上述是系统级DFT设计设备的示例,并不构成对系统级DFT设计设备的限定,可以包括比上述更多的部件,或者组合某些部件,或者不同的部件,例如所述系统级DFT设计设备还可以包括输入输出设备、网络接入设备、总线等。

[0077] 所称处理器可以是中央处理单元(CentralProcessingUnit,CPU),还可以是其他通用处理器、数字信号处理器(DigitalSignalProcessor,DSP)、专用集成电路(ApplicationSpecificIntegratedCircuit,ASIC)、现成可编程门阵列(Field-ProgrammableGateArray,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者所述处理器也可以是任何常规的处理器等,所述处理器是所述系统级DFT设计的控制中心,利用各种接口和线路连接整个系统级DFT设计设备的各个部分。

[0078] 所述存储器可用于存储所述计算机程序和/或模块,所述处理器通过运行或执行存储在所述存储器内的计算机程序和/或模块,以及调用存储在存储器内的数据,实现所述系统级DFT设计设备的各种功能。

[0079] 所述存储器可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据手机的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器可以包括高速随机存取存储器,还可以包括非易失性存储器,例如硬盘、内存、插接式硬盘,智能存储卡(SmartMediaCard,SMC),安全数字(SecureDigital,SD)卡,闪存卡(FlashCard)、至少一

个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

[0080] 本发明还提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现所述的一种系统级DFT实现方法的步骤。

[0081] 所述系统级DFT设计系统集成的模块/单元如果以软件功能单元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。

[0082] 基于这样的理解,本发明实现上述系统级DFT实现方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,所述计算机程序在被处理器执行时,可实现上述系统级DFT实现方法的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或预设中间形式等。

[0083] 所述计算机可读存储介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(Read-OnlyMemory, ROM)、随机存取存储器(RandomAccessMemory, RAM)、电载波信号、电信信号以及软件分发介质等。

[0084] 需要说明的是,所述计算机可读存储介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读存储介质不包括电载波信号和电信信号。

[0085] 实施例

[0086] 下面结合实施例与附图对本发明作进一步说明:

[0087] 为了解决现有测试技术必须借助外部测试设备才能完成信息处理微系统的测试和评价的技术问题,本实施例提供了一种系统级DFT实现方法,具体包括:

[0088] 第一,可测试性系统架构设计:

[0089] 信息处理系统微系统一般性架构及结构,如图1所示,信息处理系统架构相对统一,一般为“处理器+FPGA”为核心的架构,通过增加处理器数量、提升处理器核心数和主频达到系统性能扩展的目的,通过FPGA的可重复编程和强时序约束特性实现系统差异化定制。为保证处理器和FPGA的性能,通常为处理器和FPGA配置独立的RAM存储器和ROM存储器作为程序和数据存储使用。

[0090] 如图2可以看出,整个封装体包括封装基板100、热沉101、第一TSV中介层104、第二TSV中介层107、焊球108、微凸点109、FPGA200、第一信号209、处理器210、第一存储器212、第二存储器222、第二信号219和第三信号230。其中,处理器210和其私有第一存储器212集成在一个第一TSV中介层104上,第二存储器212通过微凸点109互连实现3D堆叠,那么,对第一存储器212的测试必须通过处理器210;同理,FPGA200和其私有第二存储器222集成在一个第二TSV中介层107上,第二存储器222通过微凸点109互连实现3D堆叠,那么,对第二存储器222的测试必须通过FPGA。第一TSV中介层104和第二TSV中介层107通过焊球108焊接在封装基板100上,通过封装基板100内第三信号230实现数据交互;处理器210和FPGA200均有IO、通讯接口、JTAG调试接口等通过TSV中介层(第一TSV中介层104上和第二TSV中介层107)、封装基板100、焊球108实现对外引出;在基板100上覆盖热沉101形成整个封装体。

[0091] 本实施例在该系统架构和典型结构下,充分利用FPGA的可重复编程、数据交换中心的特征,通过设计FPGA内运行的测试控制逻辑,实现以FPGA为核心的系统BIST控制。

[0092] 当前主流处理器均支持多接口BOOT的功能,BOOT接口通常为存储器访问接口(如并行总线、SPI、I2C等)和通讯接口(如GMAC、UART、SPI、CAN、I2C、USB、PCIe、SRI0等)。

[0093] 在微系统设计中,通常将并行总线、SPI、PCIe或SRI0作为处理器和FPGA的双向数据接口,当这些接口具备处理器BOOT功能时不需要做特殊处理,当这些原定接口不具备BOOT功能时,需要将处理器一个BOOT接口接入FPGA,以实现FPGA对处理器的程序BOOT。

[0094] 处理器的BOOT接口选择通常是通过其GPIO信号控制的,这些BOOT接口控制信号需要接入FPGA,以实现自检测试和用户程序的切换。另外,处理器往往是在外部复位信号的控制下实现上电后状态锁存和程序重新加载,其外部复位信号也需要接入FPGA,由FPGA控制其复位逻辑。

[0095] 当前主流FPGA内置多通道ADC功能,基于该ADC功能可以设计微系统模块的关键模拟量自检。设计中可选的将内部电源、测温信号接入FPGA模拟输入IO,可用于系统电源和温度自检。

[0096] 第二,自测试状态机构建:

[0097] 如图3所示,FPGA加载默认的Header代码,代码中集成ADC采集功能或者外置ADC控制功能,实现对微系统电源、温度等模拟量的转换及检测,判定当前工作状态是否符合预期,同时根据该环境参数可设定不同的工作参数,在环境自检合格的情况下,完成步骤2外部IO状态自检,通过IO高低电平确定是否进行下一步FPGA自检操作,当需要FPGA自检时,执行步骤3,首先修改FPGA配置寄存器中下次启动地址为第一份FPGA自测试向量存储起始地址,再启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载第一份FPGA自测试向量;步骤4是在FPGA加载完第一份自测试向量后,执行对FPGA部分资源的自检测,检测结果固化在FPGA配置ROM或外置专用ROM中,自检测完成后判断是否完成全部FPGA自检测向量加载,如果没有,则重复步骤3和步骤4,依次加载更多测试向量,完成对FPGA更多资源的测试覆盖。

[0098] 在完成或跳过FPGA自检后,执行步骤5对FPGA外部IO的自检,确定其状态是否为处理器、存储器及其它功能自检需求,这些IO状态和测试需求对应关系是可以自定义的。当需要处理器、存储器及其它资源的自检时,进入步骤6,首先修改FPGA配置寄存器中下次启动地址为其它功能测试向量存储起始地址,再启动IPROG命令,使FPGA完成自复位,进入重新加载流程,加载其它功能自测试FPGA代码;代码加载完成后FPGA开始执行步骤7,即控制多个独立的处理器配置接口从FPGA的ROM中读取测试向量数据,首先读取第一份处理器自测试程序,处理器完成测试向量加载后,开始自动执行该测试向量,自测试结束后返回测试结果给FPGA,FPGA将检测结果固化在FPGA配置ROM或外置专用ROM中,待全部并行执行的处理器完成当前向量自检测后,判断是否完成全部处理器自检测向量加载,如果没有,则重复步骤7至步骤9,依次加载更多处理器测试向量,完成对处理器更多资源的测试覆盖。

[0099] 待全部完成自检测试向量执行后,启动Golden段代码实现最终结果判定,以及到用户代码段的跳转。

[0100] 第三,FPGA测试向量生成及固化:

[0101] FPGA是一种特殊的器件,由于它具有可编程的特性,测试的难度和复杂程度都要远远高于ASIC的测试。由于FPGA的应用越来越广,相应的对FPGA器件的测试研究也是越来越多,根据测试需要大致可以分为两类:面向应用的测试过程(ATP)和面向制造的测试过程

(MTP),无论是这两种中的哪种测试都需要大量的测试配置文件和测试激励。但是FPGA每配置一次文件都需要数百毫秒的时间,这样把整个芯片测试完需要时间很长,所以在设计FPGA测试向量时关键要考虑更少的配置文件和更高的测试覆盖率的问题。

[0102] 本实施例中,FPGA测试向量的开发可继承FPGA单芯片测试向量,只是需要预留部分资源作为逻辑控制功能,包括结果采集、ROM编程和IPROG控制等少量状态机的资源。

[0103] 多份FPGA测试向量代码的固化采用FPGA通用MultiBoot数据流固化方式,对每个bit流进行压缩,以减小数据量,节约ROM空间,按照预设的起始地址逐个加载bit文件到MCS文件中,形成一个完整的MCS文件,再烧写到配置ROM中。

[0104] 第四,处理器测试向量生成及固化:

[0105] 本实施例中,处理器测试向量采用最小化原则生成,即将处理器相关功能模块按照最小化原则进行分割,分别实现不同功能的测试向量,减少功能之间的耦合。生成独立的功能测试向量通常是不同处理器编译环境下的特定格式数据,这些数据需要使用相关工具转换为bin文件,bin文件作为最后的固化文件使用。

[0106] 在功能模块对应测试向量bin文件的固化顺序上,通常采用从基础到独立的原则,优先固化基础功能部件bin文件,如复位测试、时钟测试等功能测试向量优先,外设接口功能测试向量置后。

[0107] 处理器测试向量bin文件的固化通常采用两种方式,第一种是基于FPGA编程软件,在生成MCS文件时依次加载不同的bin文件进去,最终形成一个MCS文件,然后固化到外部ROM里;第二种是在FPGA内实现处理器接口到外部ROM接口的控制逻辑,通过处理器代码段的执行,读取外部bin文件,通过FPGA实现的桥接逻辑实现对ROM的编程。

[0108] 本实施例提供的一种系统级DFT实现方法,相比现有技术,具有如下优势:

[0109] 本实施例采用系统架构分析、BIST测试控制逻辑的植入、测试向量实现等技术手段,开发系统级DFT功能,功能覆盖率可做到100%,设计可行性高,测试设备依赖程度低,自动化程度高,代码继承性强。不改变原有线路拓扑和负载,不增加信号线复杂度,不增加产品工艺复杂度和难度,普适性高、实现成本低。

[0110] 上述实施例仅仅是能够实现本发明技术方案的实施方式之一,本发明所要求保护的范围并不仅仅受本实施例的限制,还包括在本发明所公开的技术范围内,任何熟悉本技术领域的技术人员所容易想到的变化、替换及其他实施方式。

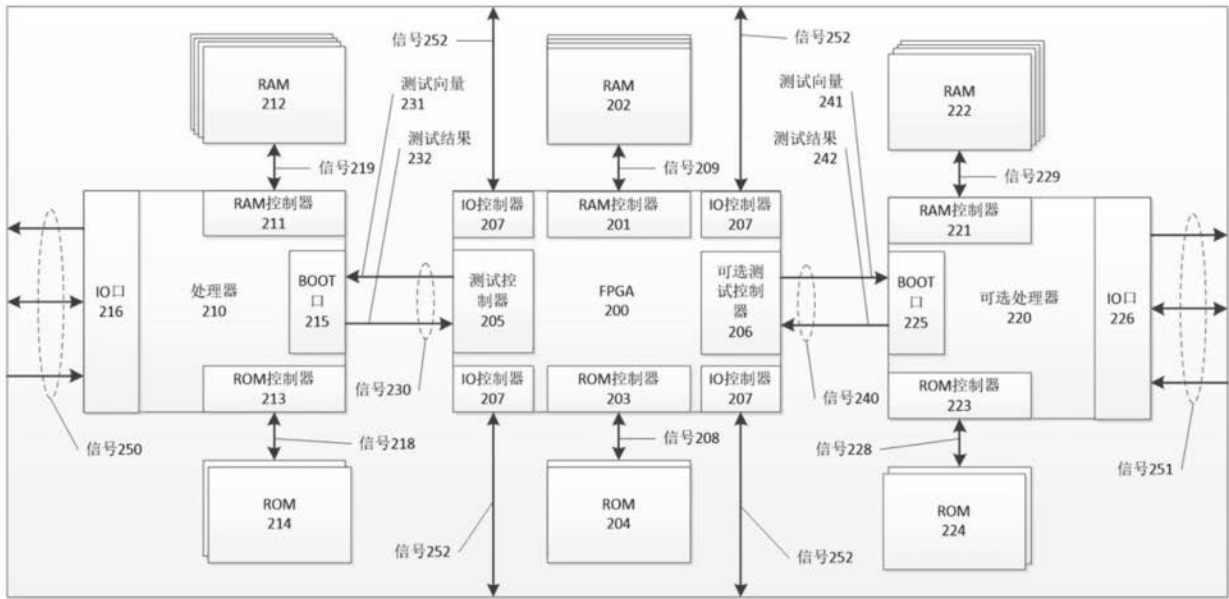


图1

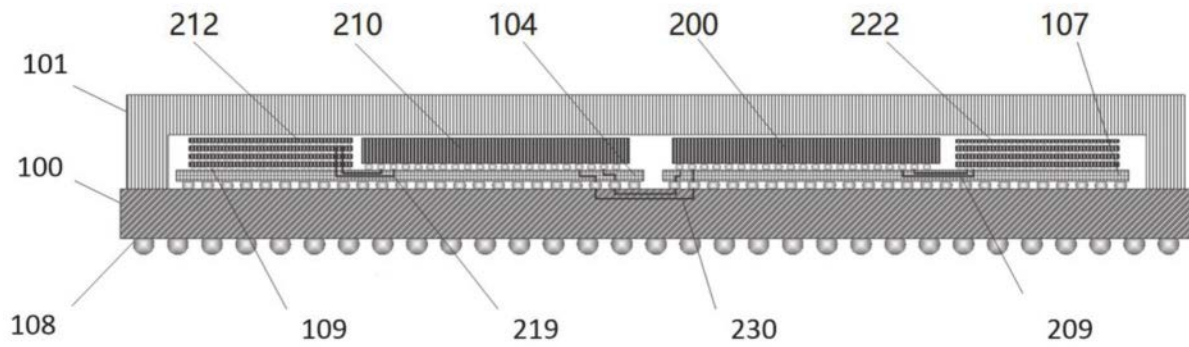


图2

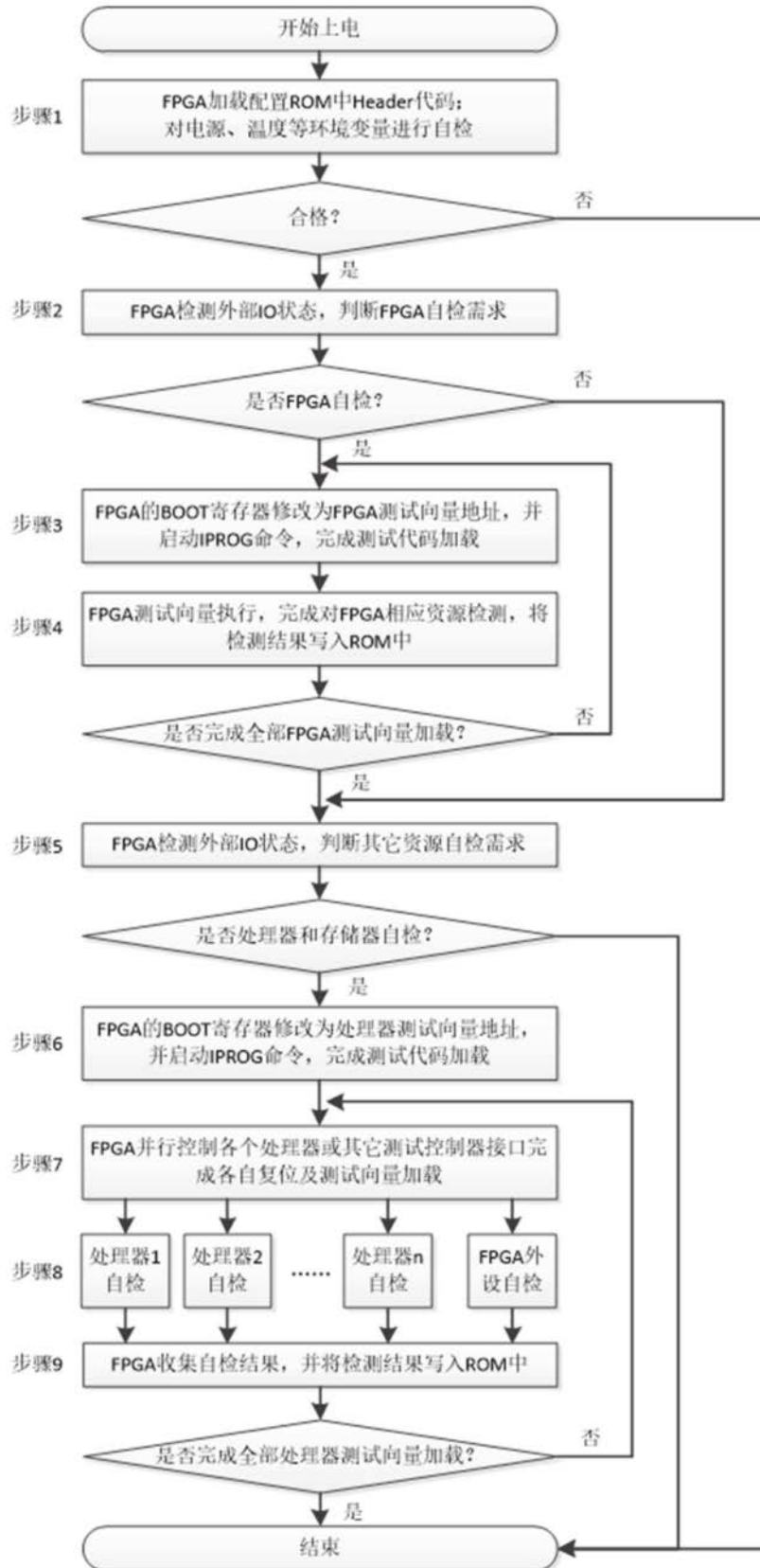


图3

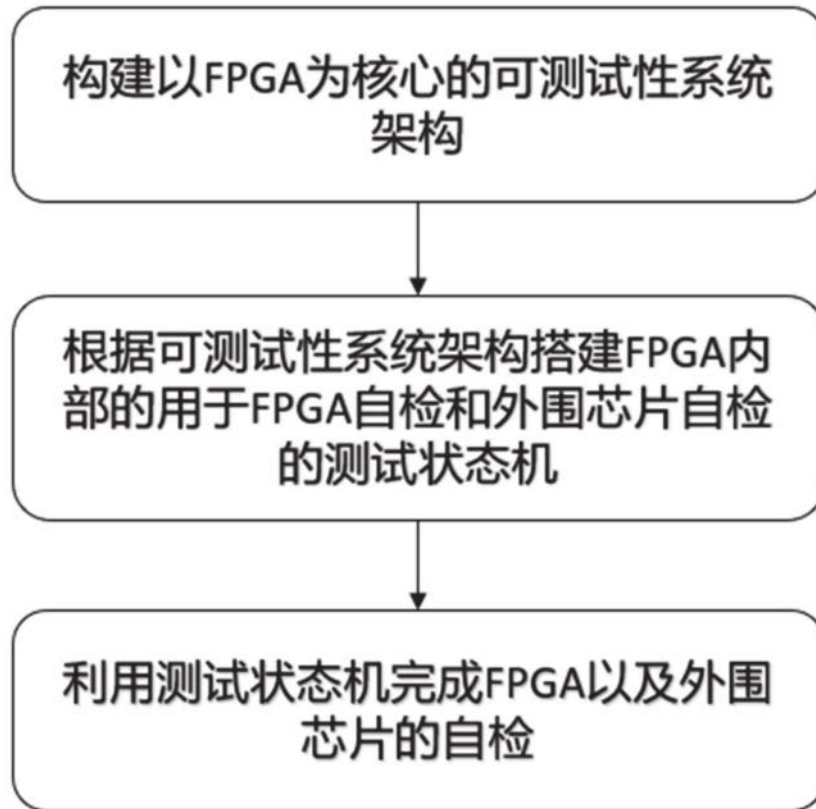


图4



图5