基于 Palladium AVIP 的 SoC 验证方案

程 涛
(哲库科技(上海)有限公司,上海 201210)

摘 要:由于片上系统芯片(System on Chip,SoC)规模越来越大,软件仿真速度在一些大的场景测试用例下已经很难满足验证计划时间的要求。现场可编程门阵列(Field Programmable Gate Array,FPGA)原型验证平台容量的限制,以及需要修改时钟树等特性导致 FPGA 平台并不适合做功耗/性能评估。基于 Emulator 平台的仿真加速以及功耗/性能评估已经成为一种趋势。可以使用 Emulator 的加速验证知识产权(Accelerated Verification Intellectual Property,AVIP)替换软件仿真用的验证知识产权(Verification Intellectual Property,VIP)来做仿真加速。以及使用高级微控制器总线结构(Advanced Micro-controller Bus Architecture,AMBA) AVIP 来模拟或者监控总线的传输,结合其他工具可以用来做功能/功耗/性能相关的验证工作,大大加速了芯片相关开发验证的进程。基于移动行业处理器接口(Mobile Industry Processor Interface,MIPI) AVIP 开发的用于验证 MIPI 或使用 MIPI 来验证内部图像处理模块的场景,达到了几十倍的加速比,大大提升了仿真速度。使用 AVIP 来模拟以及监控相关模块的行为,可以比较方便快捷地得到功耗相关数据以及性能相关数据。对芯片的功能验证,功耗/性能优化起到了至关重要的作用。

关键词:Emulator;AVIP;仿真加速;功耗分析;性能评估
中图分类号:TN402 **文献标识码**:A DOI:10.16157/j.issn.0258-7998.219803

中文引用格式:程涛. 基于 Palladium AVIP 的 SoC 验证方案[J].电子技术应用,2021,47(8):52-55.
英文引用格式:Cheng Tao. SoC verification solution based on Palladium AVIP[J]. Application of Electronic Technique,2021,47(8):52-55.

SoC verification solution based on Palladium AVIP

Cheng Tao
(ZEKU Technology(Shanghai) Co.,Ltd.,Shanghai 201210,China)

Abstract: Due to the increasing scale of the chip, the EDA simulation speed has been difficult to meet the schedule requirements in some large scene cases. At the same time, the capacity of the FPGA prototype verification platform is limited, and some clock trees need to be modified and are not suitable for power/performance evaluation. Simulation acceleration and power analysis and performance evaluation based on Emulator platform have become a trend. Based on the emulator AVIP to simulate or monitor related bus transaction can be used to do the related work of function/power/performance which can greatly accelerated the process of chip development. The test cases developed based on MIPI AVIP for verifying MIPI or using MIPI to verify the internal image processing module achieves dozens of times of acceleration ratio and greatly improves the simulation speed. As well as using AVIP to simulate and monitor the behavior of related modules, can get the netlist power data and performance data. It has made an important contribution to the functional verification and power/performance optimization of the chip.

Key words: Emulator; AVIP; simulation acceleration; power analysis; performance estimation

0 引言

本文主要聚焦于消费电子类的 SoC 芯片验证领域,为芯片验证过程中遇到的功能、功耗、性能验证方面的一些难题提供了有效的解决方案。相比传统的验证方式,基于 Palladium AVIP 的验证方案能更加快速高效地达成验证目的,为芯片按时成功流片提供了强有力的保障。

1 AVIP 简介

AVIP 是在仿真 VIP 的基础上专门针对 Emulator 平台而设计的。如果仿真用的是 Cadence 的 VIP 的话,很容易就可以切到基于 Emulator 的 AVIP 上去。基本原理是把原本跟待测设计(Design Under Test,DUT)交互的 driver、monitor、interface 等模块做了一些拆分和修改:软件侧的代理模型(proxy)提供控制接口给用户控制发包行为,并与硬件侧的 BFM 进行事务处理(transaction)级别的交互;硬件侧通过总线行为模型(Bus Function Model,BFM)来获取软件侧的发包请求并驱动 DUT。通过 Palladium(Cadence 公司的 Emulator 产品)可以大大加速硬件侧DUT 的原理,达到加速整体仿真的目的。目前 Cadence 的

AVIP 支持各种标准协议，如：MIPI、AMBA、通用串行总线(Universal Serial Bus，USB)、高速串行扩展总线标准(Peripheral Component Interconnect Express，PCIE)等。

1.1 AVIP 架构

AVIP 由如下两部分组成：

(1)硬件侧与 DUT 信号直接相连的 BFM；

(2)软件侧提供函数接口给用户开发测试平台。

软件和硬件之间底层物理交互是通过仿真加速卡(Simulation Acceleration，SA)把 Palladium 仿真加速器和服务器连接起来的，运行时通过 AVIP 软件侧的 proxy 与硬件侧的 BFM 进行 transaction 级别的通信。AVIP 架构图如图 1 所示。

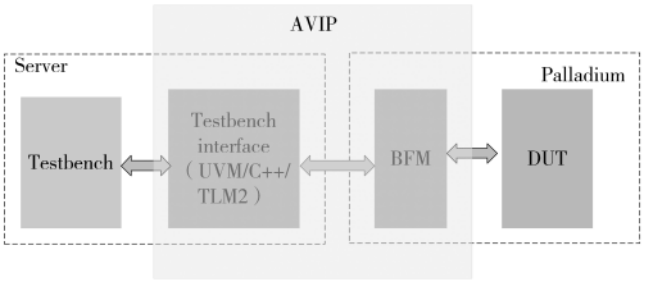

图 1 AVIP 架构图

1.2 AVIP 支持的接口

AVIP 支持多种不同的测试平台接口及使用模型，支持接口有：UVM、C++、TLM2、嵌入式等。不同的接口可以有不同的使用方式：

(1)UVM 接口一般用来作为仿真加速；

(2)C++也是用来作为仿真加速，相比 UVM 接口少了 UVM 相关的方法学的外壳；

(3)TLM2 接口可以连接虚拟平台来使用；

(4)嵌入式速度最快，需要把测试平台写成可综合的形式并综合到 Emulator 里去。

用户需要使用者根据自己平台的特性选择合适的接口和使用方式，各接口比较如表 1 所示。

表 1 各接口比较

接口	UVM	C++	TLM	嵌入式
性能	低	中	中	高
重用性	高	中	中	低
工具	仿真器/加速器	加速器(仿真器可选)	仿真器/加速器	加速器

2 基于 AVIP 的仿真加速

AVIP 的一个重要作用就是用来做仿真加速，通过加速硬件侧的 DUT，同时减少软件侧和硬件侧的交互，达到大幅加快整体仿真速度的目的。

对于一些大场景的测试用例，软件仿真速度太慢了，比如有的可能需要一两周的时间才能跑完，对于芯片验证来说很难满足验证计划的时间要求。如何在尽可能少改动原有仿真环境以及测试用例的前提下大幅提升仿真速度是一个难题。

针对不同的模块以及场景，本文将介绍 2 种不同的加速方案：UVM 仿真加速和 C++ 仿真加速方案。UVM 仿真加速方案适合设计验证(Design Verification，DV)人员，可以基于现成的 UVM 仿真环境，把 VIP 替换成 AVIP 并在 Emulator 上进行仿真加速。C++的仿真加速方案适合 SOC DV 或芯片验证(Chip Validation，CV)人员，尤其是验证 case 是基于 C 代码写的用例，优点是环境相对比较简单，没有 UVM 那一套复杂方法学，可以快速地搭建好仿真环境。并且运行速度理论上会比基于 UVM 仿真加速的方案快一些，缺点是环境复用性差一些。

2.1 基于 UVM 的仿真加速方案

2.1.1 UVM 仿真加速原理简介

使用硬件加速 DUT 的同时仍然受益于 UVM 提供的高级验证环境，如果是使用 Cadence 的 VIP 搭建的验证环境，那么可以很方便地切到 AVIP 的 UVMA 环境，只需要编译的时候增加一些编译选项和宏定义并指定两个顶层：硬件层和软件层，软件侧主要是实现 UVM 环境并运行在仿真器上。硬件侧主要是实例化 DUT 和 AVIP 的 BFM，把 AVIP 和 DUT 连接起来并运行在硬件加速器上。同时 AVIP 把原本 VIP 里与 DUT 交互的 interface 做成可综合的逻辑并放到了硬件侧，软件侧的 driver 和 monitor 可以调用硬件侧 interface 定义的 task，这样软硬件之间就可以通过 transaction 进行交互了。这样在做仿真加速时就能复用原本的 UVM 环境和测试用例，减少了重新搭建环境的工作量。

2.1.2 待测设计介绍

DUT 是一个有两个 AXI 接口的图像处理模块，一个 AXI 口是用来配置模块控制寄存器的。另一个 AXI 口是待测设计用来读取双倍数据速率存储单元(Double Data Rate，DDR)获得原始处理图像数据以及处理完数据之后把结果数据写回 DDR 通路的。

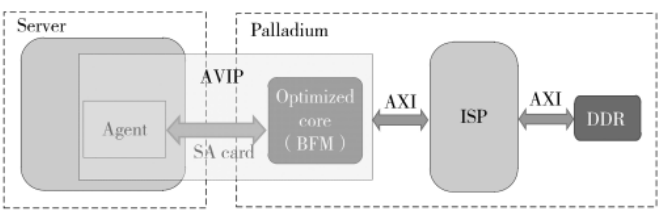

图 2 UVM 仿真加速架构图

2.1.3 仿真加速流程架构

(1)在硬件侧实例化 AVIP 的 BFM 模型并与待测设计通过信号连接起来。同时实例化一个带 AXI 口的存储模型来模拟 DDR 的功能，并与待测设计连接起来。

(2)在软件服务器侧通过调用 AVIP 提供的 UVM 接口函数来控制 AVIP 的发包行为。此处的配置信息是从算法组获得的，可以通过脚本自动生成相应的测试用例。

原始图片数据可以通过 DDR 的后门加载进去。

(3)DUT 从 DDR 读取原始数据处理完后数据通过 AXI 口写到 DDR 里面,结束后可以从 DDR 的后门导出来离线与期望的正确数据进行对比。

2.1.4 测试结果

使用这种方式的加速比可以达到十倍左右,因为有一部分非标准协议接口在,这部分使用了信号级别的通信对速度有一定影响,如果纯 AVIP 的环境理论上可以达到几十甚至上百倍的加速比。原本需要运行一周的测试用例可以十多个小时运行完,大大加速了验证速度。后续优化速度的办法是对于非标准协议或是没有 AVIP 的情况,可以对这类接口做一些优化,处理成自定义的 AVIP:主要处理 driver 和 monitor 与 DUT 交互的接口部分,把它做成可综合的逻辑并放到硬件加速器里去。

2.2 基于 C++ base 的仿真加速方案

2.2.1 基于 C++接口的仿真加速原理

Palladium AVIP 提供了运行在软件服务器侧的 proxy 模型接口,proxy 模型与硬件侧的 BFM 模型通过 transaction 发包的形式进行交互。BFM 侧则通过解析 transaction 成信号级别并驱动 DUT。

2.2.2 基于 C++仿真加速流程架构

(1)在硬件侧:实例化 MIPI AVIP 的 BFM 模型,并与待测设计通过信号连接起来。使用 Palladium 硬件来加速 DUT 侧的运行速度。

(2)在软件侧:通过调用 MIPI RX AVIP 提供的 C++接口函数,可以输入各种格式的图片数据,并控制 AVIP 按照 MIPI 协议把数据打包发送到待测设计里去。可以通过设置 AVIP 的时钟,以及插入空白行等方式控制发帧帧率,以及利用 AVIP 提供的现成函数接口控制发包的长度、类型等。

(3)结果比对:MIPI TX AVIP 可以把处理好的图片数据存成 raw 格式的文件,然后通过离线的方式进行比较,判断测试用例是成功还是失败了。

(4)AVIP 与中央处理器(Central Processing Unit,CPU)上执行的软件代码的交互:通过选取待测设计里不用的寄存器来作为 AVIP 和 DUT 的交互通道。比如 CPU 如果往某个寄存器写 0x123,testbench 侧可以通过调用在硬件描述语言(SystemVerilog,SV)里定义的直接编程接口(Direct Programming Interface,DPI)函数,来解析出这个编码,不同的编码代表不同的测试用例以及发包方式,这样就可以进行测试用例的连续测试。如此就达到了通过 CPU 软件配置寄存器来达到间接控制 AVIP 发包的目的。

2.2.3 测试结果

原来需要运行 7 个多小时的测试用例,使用基于MIPI AVIP 的仿真加速方案只需要 10 min 左右就可以运行完,加速比达到了 43 倍,大大加速了芯片验证的进度。

3 基于 MIPI AVIP 的功耗分析方案

功耗分析传统的流程是运行出波形用 Synopsys 的 PTPX 或者 Cadence 的 Joules 来算出具体的功耗值。如果是使用仿真的方式来抓取波形,对于一些大场景的测试用例面临的挑战:(1)仿真运行时间很长,有的要以周计,效率很低;(2)波形数据很大,但是功耗分析工具只能分析很短的一段波形,如何在很长的一段测试用例里准确抓取到感兴趣的那一小段波形是一个难题。

通过结合 MIPI AVIP 发包来模拟真实场景、Palladium 网表编译流程以及 Cadence 的动态功耗分析流程(Dynamic Power Analysis,DPA),HW-WTC(Hardware Weighted Toggle Count)可以解决上面提到的仿真速度慢以及难以找到感兴趣功耗波形窗口的难题。

3.1 HW-WTC 原理

HW-WTC 的基本原理是通过往硬件里增加额外逻辑统计信号翻转数,并依据.lib 文件里各个标准单元功耗信息加入相应权重,以此来统计模块所有信号的翻转数(动态功耗主要跟信号翻转率有关)。有了信号翻转数就可以通过工具分析得到某个场景下动态功耗的趋势图,有了这个趋势图就可以很方便地找到感兴趣的窗口(如峰值功耗:peak power),基于这个窗口就可以只需要抓取一小段波形提供给功耗分析工具,最终得到准确的功耗值。

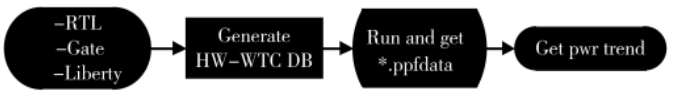

图 3 HW-WTC 流程图

3.2 解决方案

待测设计是一个图像处理芯片,通过 MIPI 接口输入输出图像数据,内部是一些处理核。

3.2.1 基于 MIPI AVIP 的功耗流程介绍

(1)编译后端提供的标准单元的.lib 文件,生成对应的 powerdb(包含各个标准单元模块功耗信息)。

(2)准备可以在 Emulator 上运行的数据库:

①在硬件侧里实例化 AVIP 的 BFM 模型,并通过 force 的方式与设计连接起来。

②编译的时候把 powerdb 和后端提供的网表文件编译成可以在加速器上运行的数据库。

(3)编写基于 C++的 MIPI AVIP 的测试平台,模拟照相机传输图片进入 DUT,并通过 CPU 配置内部图片处理模块来处理图像数据,以此来模拟真实场景下芯片的工作场景。

(4)运行完测试用例后就可以得到.ppfdata 的数据库,可以通过 xeDebug 工具查看整个测试场景下的功耗趋势图,并得到感兴趣的窗口。

(5)抓取感兴趣那段波形提供给功耗分析工具算出具体的功耗值。

3.2.2 调试技巧

最好是选取与网表相同版本的 RTL 代码先进行测试，因为网表综合布线之后很多信号的极性会发生变化，所以需要重新连接 AVIP 的信号到 DUT。在 RTL 版本上测试通过之后可以确保 RTL 和 C 代码是正确的，再上网表版本测试运行迭代次数会少很多。

3.2.3 测试结果

通过这种方案跑省去了抓取全波形的问题，通过确认感兴趣窗口以及 Emulator 的加速，如果有已经可以运行通的测试用例，顺利的话两三天就可以得到一个功耗数据，相比以前需要周计的传统的方式效率更高，结果也更加准确。

4 基于 AMBA AVIP 的性能分析方案

传统的性能分析通过查看波形效率十分低下。一些大的场景在 Emulator 上跑下来波形文件很占存储空间，同时抓波形和转波形的时间很长。仿真上运行速度又太慢，一些大的场景可能要一两周才能跑完，效率十分低下。如何能够方便快捷地得到芯片的带宽以及延迟等性能数据是个难题。

图 4 为各个主设备挂上 AMBA AVIP 性能监视器来抓取各个 master 的 transaction，并使用 Palladium 来运行真实场景。测试场景运行完直接就可以得到各个模块的发包日志。使用 SPA(System Performance Analysis)工具分析日志文件就可以得到包括带宽、延迟、outstanding 等性能结果。

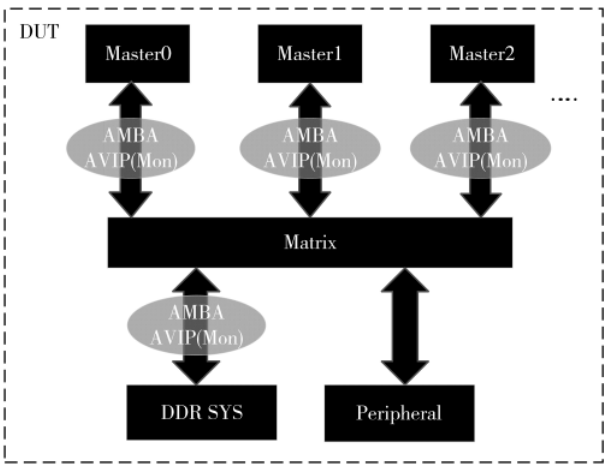

图 4 AVIP 性能监控器框图

4.1 待测设计介绍

待测设计是一个图像处理芯片，通过 MIPI 接口输入输出图像数据，内部是一些处理核。并且大部分模块的接口是 AMBA 接口。

4.2 基于 AMBA AVIP 性能方案流程架构介绍

(1)实例化 AMBA AVIP 到各个主设备口上并编译得到 Palladium 能运行的数据库。

(2)通过 CPU 执行 C 代码进行各个模块的配置，运行期望场景的测试用例，并通过 AMBA AVIP monitor 得到性能相关的日志。这些日志里面就包含了 transaction 的信息。在使用的时候可以应用上 multi GFIFO/SFIFO 技术，这样可以更加高效地抓取性能日志。因为一些大的场景日志通常也很大，multi GFIFO/SFIFO 技术可以把各个主设备的日志分开单独存放，这样使用 SPA 分析的时候可以单独分析，或者并行分析多个主设备的日志，效率会高很多。

(3)通过 SPA 工具解析这些性能日志，就可以得到包括带宽、延迟、outstanding 等性能的结果。

4.3 测试结果

最终解决了性能分析需要抓很大的波形，转波形的问题，大大节省了性能评估的时间。如果有现成的测试用例，顺利的话基本上一天就可以得到带宽、延迟、outstanding 等性能报告。相比之前抓波形转波形都要好几天加上还要人工去分析波形算性能结果，大大节省了人力和时间，对芯片架构调整或者软件代码优化提供了极大的保障。SPA 分析结果如图 5 所示。

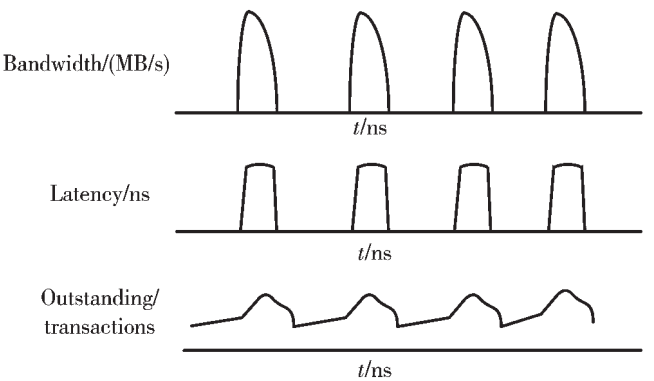

图 5 SPA 分析结果图

5 结论

结合 Palladium AVIP 来进行 SoC 芯片的验证，可以从多个维度加速芯片开发测试进度，涉及功能、功耗、性能等各个方面。解决了传统验证方案耗时耗力的问题，加快了整个芯片的验证速度，为芯片顺利流片提供了极大的保障。

参考文献

[1] Cadence Design Systems, Inc.: AXI Accelerated VIP User Guide[Z].2021.
[2] Cadence Design Systems, Inc.: CSI-2 Accelerated VIP User Guide[Z].2021.

(收稿日期:2021-06-23)

作者简介:
程涛(1987-),男,本科,高级系统工程师,主要研究方向:SoC emulation 验证。

扫码下载电子文档

版权声明