

面向 SoC 的软硬件协同验证平台设计

鲍 华¹, 洪 一^{1,2}, 郭二辉¹

(1. 华东电子工程研究所, 合肥 230031; 2. 安徽大学计算机科学与技术学院, 合肥 230039)

摘 要: 针对 SoC 设计验证的实际需求, 介绍一种面向 SoC 设计的软硬件协同验证平台。平台中软硬件模型分别在不同环境下运行, 通过网络实现信息交互。硬件用硬件描述语言实现对系统事务级、RTL 级的建模, 软件用高级编程语言来编写, 使用指令集仿真器完成对硬件的仿真。仿真过程使用不同的进程并行进行, 应用进程间通信方式实现仿真器之间的信息交互。

关键词: 协同验证; Socket 接口; 数据传输

Design of HW/SW Co-verification Platform for SoC

BAO Hua¹, HONG Yi^{1,2}, GUO Er-hui¹

(1. East China Research Institute of Electronic Engineering, Hefei 230031;

2. School of Computer Science and Technology, Anhui University, Hefei 230039)

【Abstract】 With respect to the software and hardware verification requirement of SoC design, a HW/SW co-verification platform is presented. On the platform, HW/SW model is run in different circumstance and information transform is realized through network. The hardware uses HDL to construct HW model, and it implements model of transaction and RTL level. The software uses high level program language accomplished and the simulator based ISS implementation simulation of the hardware. The simulate process executes parallel in different progress and communication through IPC.

【Key words】 co-verification; Socket interface; data transfer

1 概述

随着以 IP(Intellectual Property)核复用为核心的设计技术的出现, 集成电路(Integrated Circuit, IC)应用设计已经进入 SoC(System on a Chip)时代, SoC 是一种高度集成的嵌入式片上系统。芯片设计中的任何缺陷都会导致整个芯片的设计失败, 因此, 在流片之前, 必须对芯片的系统功能实行验证。伴随芯片复杂度的不断上升, 芯片验证工作的复杂度和工作量呈现指数上升, 对验证的依赖与要求也越来越高^[1]。作为设计过程的一部分, 验证起着越来越关键的作用, 验证的效率和可靠性往往决定了项目的成败^[2]。对比传统的验证方法, 协同验证技术使得软件设计工程师能够在设计早期进行调试、验证, 缩小软/硬件的整合时间, 可以为硬件工程师提供一套非常接近现实的激励集, 以提高验证的质量, 并缩短产品的上市时间^[3]。

本文在基于指令集仿真器(Instruction Set Simulation, ISS)仿真方法的基础上, 设计了一种基于 SystemC 和 ISS 的软硬件协同验证平台。协同验证平台由硬件执行环境和软件执行环境组成, 通过事件、命令和机制实现 2 个环境的信息交互。平台对硬件部分分别进行事务级(Transaction Level, TL)和寄存器传输级(Register Transferring Level, RTL)仿真, 对仿真速度、软硬件及验证系统架构实现了优化, 可以快速、高效地实现协同验证。

平台中软硬件执行环境之间利用网络传输数据, 并分别在客户端和服务端所处的机器上进行仿真, 实现数据处理。不同数据处理进程并行进行并通过进程间通信实现信息交互, 能有效提高验证效率。

2 软硬件协同验证平台组成结构

2.1 传统的协同验证系统

传统的协同验证系统由一个硬件执行环境和一个软件验证环境组成, 通过事件和命令在 2 个环境中进行控制和信息交互, 如图 1 所示^[3-4]。硬件仿真通过运行在工作站上的软件程序模拟, 通过设定的通信接口与软件执行模块交互。软件执行模块用于产生总线周期的序列, 序列被转换成信号事件或命令集后用于驱动对应的硬件执行命令; 同时, 对硬件的总线周期响应进行采样。软/硬件仿真使用的是独立的进程, 一般采用进程间通信(Inter Process Communication, IPC)技术和总线封装器来实现软硬件仿真器之间的通信、同步和信息交互。

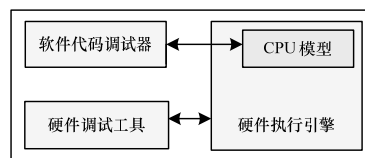


图 1 传统软硬件协同验证的基本架构

在协同验证中, 软硬件部分都是通过建模的方式实现。硬件建模方式有如下几种^[3-4]: 采用 FPGA 原型或模拟系统; 采用硬件描述语言(Hard Description Language, HDL)及逻辑

作者简介: 鲍 华(1978—), 男, 助理工程师、硕士, 主研方向: ASIC 及 SoC 芯片的设计和验证; 洪 一, 研究员、硕士、博士生导师; 郭二辉, 工程师

收稿日期: 2008-10-15 **E-mail:** baohua@mail.ustc.edu.cn

仿真器；采用高级程序语言(如 C/C++)编写的行为模型。软件可通过以下几种方式实现：采用 ISS 在 CPU 上执行；通过编译在运行验证程序的主机上运行。目前，ISS 和 RTL HDL 相结合的应用方式在协同验证中获得了广泛运用。

传统验证系统一个较为显著的局限是 IPC 通信会和主机系统的其他通信发生冲突，造成仿真性能的瓶颈。其次，总线封装器和 ISS 之间的接口是私有的，当新的内核被集成到协同验证系统中时需要修改 ISS 以支持验证系统的 IPC 原语，从而增加了复杂度^[4]。另外，验证的抽象层次低，存在精度高但速度慢和效率低的问题^[5]。

2.2 软硬件协同验证平台组成

软硬件协同验证平台在传统验证系统的基础上引入 SystemC 实现建模，并利用网络传输软件和硬件执行环境之间的数据，实现信息交互，如图 2 所示。

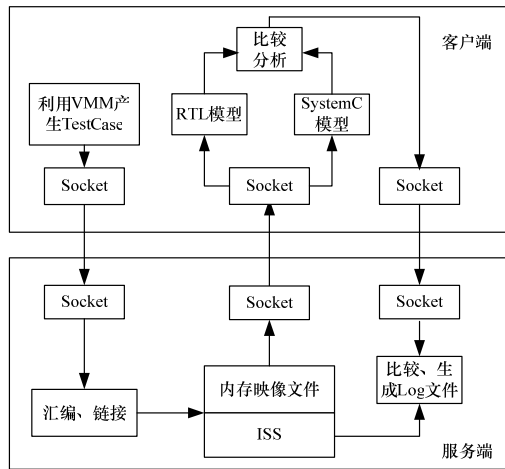


图 2 软硬件协同验证平台示意图

软硬件协同验证平台按照功能可以分为 4 个部分：(1)基于硬件描述语言编写的执行程序，应用 VMM 方法产生软件模型，即 ISS 需要的测试汇编文件集(TestCase)；(2)利用硬件描述语言编写的 RTL 级模型，在加载软件模型部分发送过来的内存映像文件后生成 RTL 级仿真结果；(3)利用 SystemC 语言编写的仿真模型，在加载内存映像文件后生成 SystemC 仿真结果；(4)ISS 仿真模型，加载测试汇编文件集(TestCase)后生成 ISS 仿真结果。

软硬件模型与汇编文件集生成程序分别在不同的机器上运行，软/硬件之间通过网络以 Socket 接口方式实现数据传输和信息交互。按照网络通信可将平台分为客户端和服务端，客户端实现测试向量(TestCase)文件的产生，服务端对接收到的测试向量进行编译，产生验证结果。客户端将产生的测试向量发送到服务端，服务端编译处理后将产生的结果反馈给客户端，客户端依据接收的反馈数据，实行比较与分析。客户端包括 3 个部分：(1)测试向量的产生，硬件执行程序应用 VMM 方法产生测试需要的测试向量；(2)寄存器传输级建模与仿真，RTL 仿真器加载测试向量生成 RTL 测试结果；(3)事务级建模与仿真，SystemC 模型加载测试向量生成测试结果，并与 RTL 模型结果实行第 1 次比较分析。

服务端主要有 2 个作用：(1)对客户端生成的测试向量文件进行汇编、链接，生成客户端软硬件模型需要的机器码；(2)利用 ISS 实现对处理器的建模，依据测试向量生成自己的结果，并与客户端的 RTL/SystemC 模型结果进行第 2 次比较分析，以增加验证的准确性，生成分析结果日记文件。

2.3 软硬件建模方式

2.3.1 硬件建模

平台的硬件建模包括事务级建模和 RTL 级建模。事务级是指在建模和仿真系统中，2 个部件之间的一次数据或事件的传输，RTL 级建模即寄存器传输级建模。事务级建模利用 SystemC 语言实现，建立相应的通信机制和通信抽象。事务级建模在开发的较早阶段进行验证，可以提供非常快速的仿真速度；同时，可以检验系统架构的合理性。RTL 级模型用 HDL 语言实现，在验证系统中与事务级模型并行在 VCS 环境下进行仿真。通过比较 RTL 级与事务级的仿真结果，可以检验和优化硬件的系统架构设计。

2.3.2 软件建模

软件模型主要是利用 ISS 仿真器模拟微处理器的行为，接收测试向量后对其编译处理后生成机器码，并对机器码进行解码和加载。软件模型通过网络与硬件仿真器进行信息交互，实现软硬件协同验证。ISS 模型比较复杂，可以采用特定处理器厂商提供的模型，也可以用 C/C++ 等高级语言模拟实现，协同仿真模型，如图 3 所示。

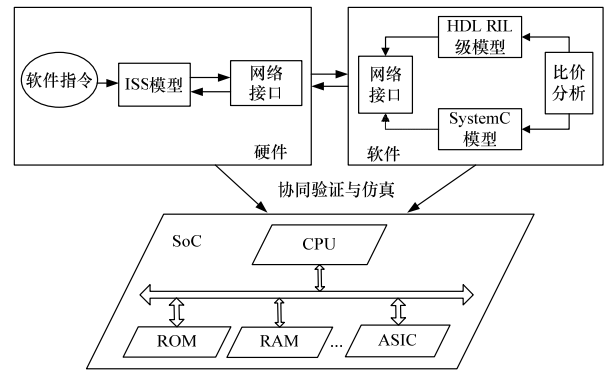


图 3 协同仿真模型

3 软硬件协同验证平台的实现

3.1 平台间数据传输方式

验证平台服务端和客户端的通信采用基于 TCP/IP 协议的 Socket 接口技术进行通信。客户端和服务端之间利用 UDP 方式收/发请求和确认信息，利用 TCP 方式传输报文。客户端/服务端通过设定的 UDP 端口(Q_SEND/ Q_ACK)侦听/接收 UDP 传输的请求/确认信息，利用设定的 TCP 端口(D_SEND/D_RECV)接收/发送报文，服务器和客户端的通信以数据包的方式实现。客户端/服务端的端口使用如表 1 所示，C 表示 Client(客户端)，S 表示 Sever(服务端)。

表 1 平台中端口的使用

客户端/服务端	端口	控制	传输路径	收/发方式
Client	Q_REQ	Require	C to S	UDP
Server	Q_ACK	ACK	S to C	UDP
Client	D_SEND	Send	C to S	TCP
Server	D_RECV	Receive	S to C	TCP

3.2 验证平台的验证流程

测试流程由客户端发送请求开始信号 Q_REQ 开始，服务端接收到 Q_REQ 后激活本机上的执行程序，并返回一个确认信号 Q_ACK 给客户端。客户端接收到确认的应答信号后激活硬件部分的执行程序，产生测试需要的测试向量，硬件执行程序可以使用 Verilog/VHDL 等硬件描述语言实现。客户端通过 Socket 接口方式将测试向量发送给服务端。服务端将测试向量的源文件编译、链接后，经过软件模型生成结

果,并提取机器码传输给客户端。客户端通过 Socket 接口接收机器码并置入 RTL 模型和 SystemC 模型,2 个模型分别产生各自的结果,并实现第 1 次比对。比对正确后将一致的结果传输到服务端的软件测试接口与软件模型生成的结果实现第 2 次比对,该次比对验证的结果被记录入 Log 文件用于核查。图 4 给出验证平台的软硬件模型生成结果的例子。

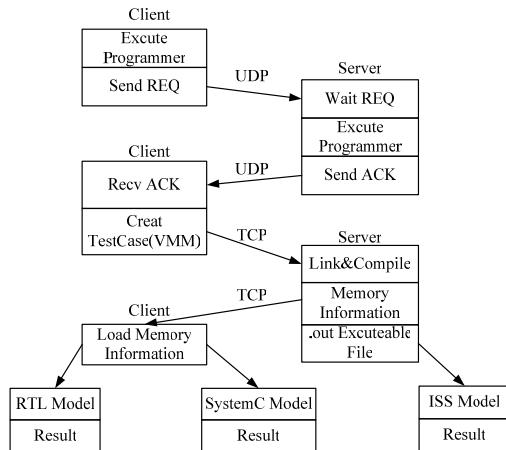


图 4 软硬件验证过程的具体实现

4 运行与测试

为对验证平台的性能做一个评估,采用 2 台主频为 1.8 GHz 的计算机分别做客户端与服务端,客户端模型在 VCS 环境下运行,服务端的仿真模型用高级编程语言 C++编写并在 Linux 环境下运行。结果显示,验证平台可以稳定地进行数据测试,自动化程度较高,数据传输率可以达到 8 MB/s,能够满足大批量数据容量 SoC 软硬件协同验证的需求。

5 结束语

设计验证是 SoC 设计的关键技术之一,贯穿整个 SoC 设计过程。随着 SoC 技术的发展,软硬件协同验证的效率和正确性对整个 SoC 设计的影响也越来越大。相比较过去常用的将软硬件仿真环境放在一个系统中,实行单一的进程处理方法不同,本文将软硬件的仿真过程分离开来,使用 Socket 来实现软硬件的通信和信息交互,更加接近实际情况。在该软硬件协同验证平台中软硬件仿真分别处于不同的环境下,仿真过程可以并行执行,仿真速度更快;同时,在不同的机器上实现软硬件仿真,可以提高数据验证的容量,效率更高。在该平台中,软件用编程语言实现,硬件用硬件描述语言来建模,在系统级进行建模,可以同时验证系统架构的正确性,以优化系统结构。

参考文献

- [1] Hopes T. Hardware/Software Co-verification: An IP Vendors Viewpoint[C]//Proc. of the International Conference on Computer Design. Austin, TX, USA: [s. n.], 1998: 242-246.
- [2] Rashinkar P, Paterson P, Singh L. System on a Chip Verification Methodology and Techniques[M]. [S. l.]: America Cadence Design Systems, Inc., 2002.
- [3] 辛 强. 一种基于 ISS 的软硬件协同验证环境[J]. 现代电子技术, 2007, 31(8): 151-153.
- [4] 张洵颖, 龚龙庆. 集成 ISS 的 SystemC 内核的协同验证环境设计[J]. 微电子学与计算机, 2006, 23(7): 53-57.
- [5] 张 奇, 曹 阳, 李栋娜, 等. 基于 SystemC 的 SoC 行为级软硬件协同设计[J]. 计算机工程, 2005, 31(19): 217-219.

编辑 顾逸斐

(上接第 270 页)

考虑到数字地球平台的基本特性,运动模型仅考虑直线和圆周运动 2 种基本运动形式,其他复杂运动形式通过这 2 种形式构建^[3]。运动轨迹表现模型则根据轨迹表现方式的不同,分成点模型、线模型和矩形模型 3 种。

4 基于想定驱动的虚拟海战场环境仿真

建立基于数字地球的虚拟海战场环境的目的在于辅助作战指挥员更快速直观地理解和分析战场态势。目前常用的方法是利用拟制的想定作为整个战场态势推演的基础^[4]。在其基础上,建立基于数字地球的全局一致的虚拟海战场环境视图,同时指挥员还可以通过相应的想定编辑接口和模型交互接口实现对战场态势的推演控制和战场实体操纵,从而有效地达到通视、监控和分析调整战场态势的目的,如图 6 所示。

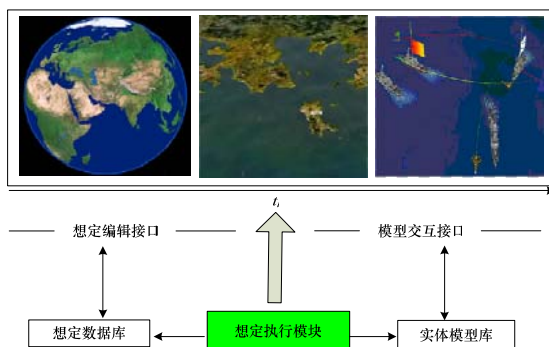


图 6 想定驱动的虚拟海战场仿真

5 结束语

本文给出了基于数字地球的虚拟海战场环境仿真思路和技术框架,实验证明其是传统数字海图和二维军标的有效补充,能为作战指挥员提供良好的态势理解和分析平台。但本文的讨论仅考虑了海战场中水面和水上各环境要素,在实际的海战场环境建设中可能还需要考虑水下地形地貌、涌浪、水声等重要信息对水面、水上以及水下等目标实体的影响,这也是本文下一步的研究方向和目标。

参考文献

- [1] Li Sujun, Yang Bing, Wu Lingda. Efficient View-dependent Modeling and Rendering of Large-scale Ocean Wave Based on Digital Earth[J]. International Journal of Computer Science and Network Security, 2007, 7(5): 118-126.
- [2] Yang Xudong, Pi Xuexian, Zeng Liang, et al. GPU-based Real-time Simulation and Rendering of Unbounded Ocean Surface[C]// Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics. Hong Kong, China: [s. n.], 2005.
- [3] 王兴申, 田康生. 基于运动方程的目标航迹模拟方法[J]. 现代电子技术, 2006, 29(5): 4-6.
- [4] 方胜良. 信息对抗效能评估通用仿真框架研究[J]. 军事运筹与系统工程, 2005, 19(3): 57-60.

编辑 张 帆