

基于 VMM 方法学的系统级软硬件协同仿真验证

章林柯¹, 王力², 王艳武³

(1. 海军工程大学 振动与噪声研究所, 湖北 武汉 430033;

2. 东南大学 集成电路学院, 江苏 南京 210096;

3. 92601 部队, 广东 湛江 524009)

摘要: 针对一款高性能复杂 SoC 芯片的设计, 提出了一种新的软硬件协同仿真验证方案。通过比较仿真环境中软硬件间通信的各种实现方式, 构建了一种新的符合 VMM 标准的验证平台。同时为加快覆盖率的收敛速度, 给出了随机激励约束的优化方法。实践表明, 新的约束和仿真方式使覆盖率收敛速度提高数倍, 验证效率显著提高。

关键词: VMM 方法学; 软硬件协同验证; 验证平台; 覆盖率; SoC

中图分类号: TN402

文献标识码: A

文章编号: 1674-7720(2011)12-0081-04

VMM-based system-level software and hardware co-simulation verification

Zhang Linke¹, Wang Li², Wang Yanwu³

(1. Institute of Noise and Vibration, Naval University of Engineering, Wuhan 430033, China;

2. College of Integrated Circuits, Southeast University, Nanjing 210096, China;

3. 92601 Unit, Zhanjiang 524009, China)

Abstract: To aim at a high-performance complex SoC chip design, the paper proposes a new hardware and software co-simulation verification. By comparing various hardware and software communication implementations for the simulation environment, a new platform to meet the VMM verification standards was built. At the same time to speed up the convergence rate, the optimization method for random constraints was given. Practice showed that with the new constraints simulation approach the convergence rate was increased by several times and the new scheme significantly improved the verification efficiency.

Key words: VMM; hardware and software co-simulation verification; verification platform; coverage; SoC

为适应日益复杂的系统芯片 SoC(System on Chip)设计, 新一代芯片设计和验证语言 SystemVerilog 应运而生, 并在 2005 年 11 月被美国电气和电子工程师协会 (IEEE) 批准为新标准^[1]。在此基础上, Synopsys 公司和 ARM 公司联合推出了 VMM 验证方法学^[2,3]。运用 VMM 方法学提出的规则和标准函数库, 可以快速搭建功能强大的验证平台。通过受约束的随机激励, 并以覆盖率为指导, 可以快速完成系统功能验证, 显著提高验证效率。

文章以实际工程项目为背景, 构建了一种符合 VMM 方法学标准的系统级软硬件协同验证平台。同时通过对实验数据的分析, 提出了用于优化随机激励约束的方法。

1 系统级软硬件协同仿真验证平台

一般仿真验证 SoC 的策略可分为 3 个步骤: 模块验证、集成验证和系统验证^[2,4]。其中系统验证平台中包含处理器 CPU 或 DSP^[2], 并且采用软硬件协同验证方法。因此系统级软硬件验证比其他形式的验证更贴近真实环境, 在仿真中可以观察到软硬件运行的所有情况, 这样可以快速有效地定位问题并进行系统性能综合分析。但是系统级软硬件协同仿真验证也存在验证平台搭建周期长、编译仿真时间长和资源消耗大的问题。针对这些问题可以采用 VMM 方法学推荐的方法和标准函数库, 并采用成熟的 VIP (Verification Intellectual Property), 快速搭建验证平台。同时通过一次编译多次仿真的验证

技术与方法 Technique and Method

方式,以提高验证效率,缩短验证周期。

图1所示是一种符合VMM标准的系统级软硬件协同仿真验证平台。在这里,硬件指RTL(Register Transfer Lever)设计,包括ARM核处理器模型。软件指用汇编语言和C语言编写的程序,通过编译加载到仿真环境中由ARM核处理器模型执行^[5]。验证平台由SystemVerilog语言实现,在RTL级仿真验证。

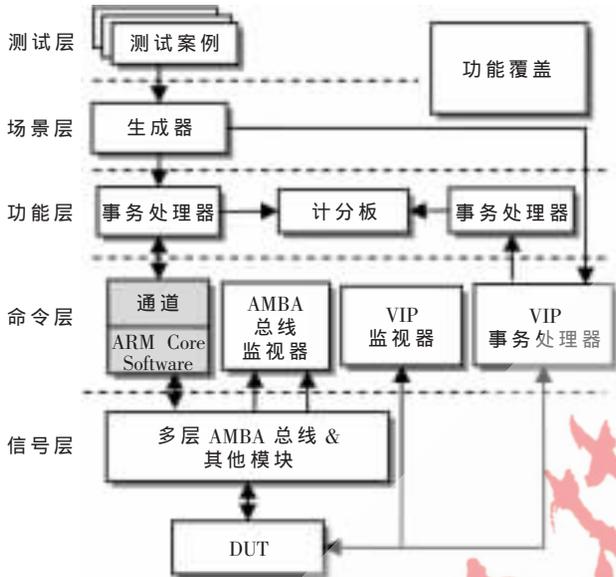


图1 系统级软硬件协同仿真验证平台

1.1 验证平台的架构

图1所示的验证平台符合可重用性^[2,6],即不同的DUT(Design Under Test)模块可以共用同一个验证平台。验证平台结构被模块化和层次化。每个模块的功能和层与层之间的接口被明确定义,这使得验证平台十分灵活。

1.1.1 测试层

测试层主要是编写各种测试案例(Testcase),包括修改生成器的约束,定义新的随机场景,同步不同事务处理器和创建定向激励^[2]。通常,在仿真的进行中要不断查看搜集的覆盖率,并以此来修改随机激励的约束条件,或者针对难以覆盖的边角情况编写定向激励。适当的约束条件可提高覆盖率收敛速度。

1.1.2 场景层

场景层由生成器(Generator)产生可控且同步的事务。生成器根据测试层提供的静态变量和约束条件,随机生成不同的配置信息和数据。配置信息通过功能层的事务处理器传递给命令层的软件程序,用于配置DUT。同时配置信息还要传到VIP事务处理器用于配置VIP模块。

1.1.3 功能层

功能层将场景层随机生成的配置信息和数据通过事先约定的通道传递给命令层的软件程序,另外检测VIP事务处理器传回的数据,送给计分板(Scoreboard),实现验证平台的自动检查功能。

1.1.4 命令层

命令层包括控制ARM核运行的软件程序(ARM Core Software,图1中带纹理方框),需要接收功能层通过通道(Channel)传递过来的配置信息和数据,同时还要配置和驱动DUT完成数据传输。VIP事务处理器(VIP Transfer)是用于验证DUT的成熟验证IP模块。VIP监视器(VIP Monitor)属于VIP模块,用于监视DUT与外部通信的接口信号。命令层还包含一个或多个AMBA总线监视器,属于成熟的VIP模块,用于监视多层AMBA总线(Multi-layer AMBA Bus)^[7]。

1.1.5 信号层

在信号层,例化了所有SoC模块,实现系统级的功能验证。除AMBA总线外,主要包括ARM核、片上RAM、中断控制模块、DMA控制模块以及其他模块。

1.1.6 功能覆盖率

功能覆盖率可以从验证平台的其他模块中收集,一般没有特定的限定。

1.2 验证平台中软硬件通信方式

上面所介绍的验证平台与其他验证平台最大的区别就是引入了软件程序,构建该验证平台的关键是软件与硬件之间的通信。一般软件是用汇编和C语言编写的,硬件与验证平台分别是用可综合Verilog语言和SystemVerilog语言编写的。在这里规定用软件程序与SV(SystemVerilog)之间的通信表示软件程序与硬件之间的通信,本文介绍以下三种软硬件通信方式。

- (1)软件向总线未用地址写数据实现软件程序向SV的单向通信;
- (2)通过SoC上的具有通信功能的模块,实现软件程序与SV之间的通信;
- (3)通过片内RAM,实现软件程序与SV间的双向通信。

在实际使用过程中,通过第一种方法输出软件程序运行过程中的信息,通过第三种方法可以作为软硬件间信息通信的双向通道。图1所示的软硬件协同仿真验证平台就是通过第三种方式实现软硬件双向通信的,同时也用到了第一种方式,输出软件运行过程中的信息。第二种方式可以视情况使用。

图2所示是一种系统级软硬件协同仿真验证平台的结构,虚线中为SoC的系统架构。为满足验证需求,在SoC系统架构中增添了两个模块:一个是Print Tube(PT)模块,用来实现软件仿真时的信息输出。当软件需要输出信息时,向PT模块写数据,PT模块会调用\$display系统函数,显示传输过来的信息。另一个是RAM(Random Access Memory)模块,作为软件和SV的信息交互的通道。RAM模块有两个总线接口,一个链接到SoC系统总线上,软件可通过其读写RAM模块;另一个接口与验证平台中的BFM(Bus Function Model)模块链接,如

技术与方法 Technique and Method

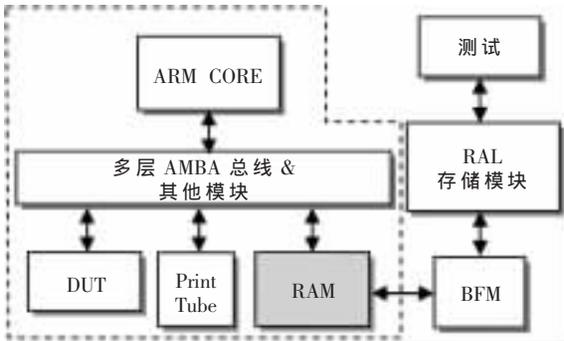


图2 系统级软硬件协同仿真验证平台的结构

AHB VIP Master^[8]。同时在验证平台中使用RAL(Register Abstraction Layer)^[9]方法,将RAM模块抽象为RAL模型。这样处理后验证平台对RAM模块的访问更简洁方便。

为验证SoC系统上某个模块的功能,需要编写两套激励:C语言编写的软件程序和SV语言编写的激励。在仿真时,两套激励需要同步。可以通过监测RAM模块中某个寄存器的变化实现同步。

在搭建图1所示的系统级软硬件协同仿真验证平台时,由于大量使用了VIP,同时开发了可重用的PT模块和SRAM模块,这使得验证平台的搭建工作可以在一周内完成,迅速提高了验证效率。

2 受约束的随机激励与功能覆盖率组

搭建好基于VMM标准的验证平台后,要发挥平台的作用,关键是要设计好受约束的随机激励。

一种分类树(Classification Trees)^[10]的方法可以有效地将待测模块(DUT)的各种功能配置情况转换成System-Verilog^[11-3,11]的约束语句。但是一般在编写受约束随机激励前,需要通过一些定向激励确定验证平台工作正常,以及配置参数有效取值。

如下程序是验证项目中为验证SPI模块DMA传输方式编写的约束条件。在经过试验性的定向测试和分类树处理后,形成SystemVerilog的约束语句。

程序1:

```
constraint c_ssi_dma {
  ch      dist { SSI1 := 1, SSI2 := 0, SSI3 := 0 };
  ctrlr0_spc dist { 0 := 1, 1 := 0, 2 := 0, 3 := 1 };
  (burst_size == 0) -> (dmatdtr inside {[0:20]})
    &&(dmardlr == 0)&&(datas.size() inside {[79:80]});
  (burst_size == 1) -> (dmatdtr inside {[0:17]})
    &&(dmardlr == 3)&&(datas.size() inside {[77:80]});
  (burst_size == 2) -> (dmatdtr inside {[0:13]})
    &&(dmardlr == 7)&&(datas.size() inside {[73:80]});
  (burst_size == 3) -> (dmatdtr inside {[0:5]})
    &&(dmardlr == 15)&&(datas.size() inside {[65:80]});
  .....
  .....
```

```
datas.size() <= 80;
for_block_ts == (80-datas.size());
}
```

在编写完随机约束条件的同时还要编写相应的覆盖率组(covergroup)^[11-3,8],用于自动收集仿真过程中功能覆盖情况,指导仿真。如下程序是针对SPI模块DMA传输方式编写的功能覆盖率组。

程序2:

```
covergroup covport_ssi_dma;
  coverpoint ch {bins CH[]={1:3}};
  ctrlr0_spc_c:coverpoint ctrlr0_spc {
    bins spc_00={0};
    bins spc_11={3};
  }
  coverpoint burst_size {
    bins burst_size_1={0};
    bins burst_size_4={1};
    bins burst_size_8={2};
    bins burst_size_16={3};
  }
  dmatdtr_c:coverpoint dmatdtr {
    .....
    .....
    .....
  }
  cross ch, ctrlr0_spc_c, burst_size, dmatdtr_c, dmardlr_c,
    for_block_ts, ctrlr0_dfl_c {
    ignore_bins
    burst_t1=binsof(burst_size)intersect{0}&&
      binsof(dmatdtr_c)intersect{[21:$]};
    ignore_bins
    burst_t4=binsof(burst_size)intersect{1}&&
      binsof(dmatdtr_c)intersect{[17:$]};
    .....
  }
endgroup
```

3 验证过程与验证结果

3.1 验证过程

仿真所采用的软件为Synopsys公司的VCS-MX-C2009.06。它自带有VMM标准库函数,支持一次编译多次仿真。由于如图1所示的软硬件协同仿真验证平台的通用性,可以将所有测试案例一次编译后,分别仿真。不同的测试案例代码写到VMM的宏'vmm_test_begin()'与'vmm_test_end()'之间,并在验证平台的程序(program)中通过类(class)vmm_test_registry的方法run()来调用。仿真过程如图3所示。

3.2 验证结果

图4是由上面程序2所定义覆盖率组收集的覆盖率数据。每次仿真中,随机产生10次随机场景。这样做可以减少重复运行仿真的次数,提高仿真速度。

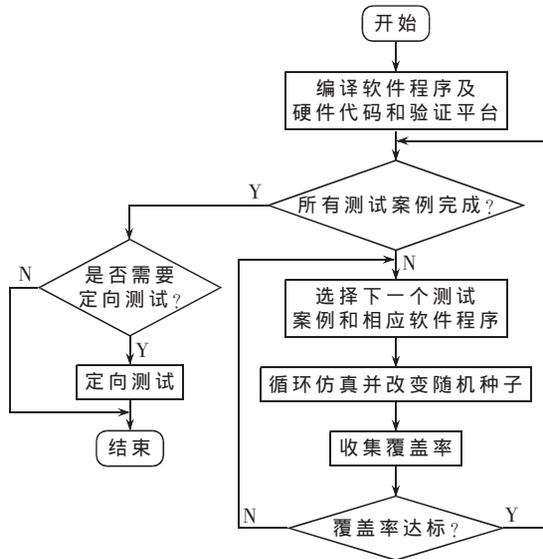


图3 验证过程

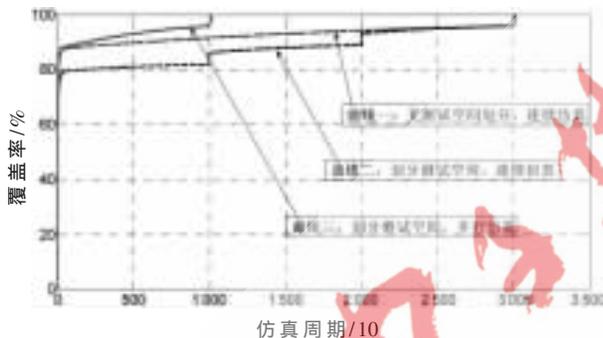


图4 覆盖率-仿真周期变化曲线

从仿真结果看到:(1)覆盖率在仿真初始阶段增长很快,随后逐渐趋缓;(2)划分测试空间后的连续仿真,与未划分测试空间的连续仿真最终仿真时间相同;(3)划分测试空间后的并行仿真,覆盖率收敛速度最快,可以使覆盖率收敛速度提高近3倍。

另外由结果(1)可知,在仿真过程中,通过负反馈机制修改随机变量的约束条件,排除已测试过的测试案例,可以使每次仿真的覆盖率保持很高的增长速度^[12]。但修改约束条件后需要重新编译。

针对实际项目需要,文章介绍了一种符合VMM标准的系统级软硬件协同仿真验证平台,讨论了验证平台中软硬之间的通信方式。给出了在此平台上所做的一个模块的具体验证,包括随机激励的约束和相应的覆盖率组。最后对仿真结果做了对比分析。结果表明,与基于SystemC语言搭建的验证平台^[13]相比,基于VMM标准并使用SystemVerilog语言,可以提高验证平台的搭建效率和可重用性。通过对测试空间的划分和仿真过程的控制,有效提高仿真效率。

参考文献

[1] IEEE standard for SystemVerilog-unified hardware design, specification, and verification language-IEEE computer

society[S].IEEE Std 1800™-2007.

- [2] BERGERON J, CERNY E, HUNTER A, et al. SystemVerilog methodology manual for SystemVerilog[M]. Springer, 2005.
- [3] CHRIS S. SystemVerilog for verification[M]. Springer, 2006.
- [4] 程刚,蔡敏.基于SystemVerilog的SoC功能验证方法研究[J].科学技术与工程, 2009,9(22):6814-6818.
- [5] 申敏,曹聪玲.基于SoC设计的软硬件协同验证技术研究[J].电子测试, 2009, 17(3):9-12.
- [6] Synopsys Inc. Introduction to design verification with VMM: A quickstart guide. Version 1.0[S]. 2007, 2008.
- [7] ARM Limited. AMBA™ Specification (Rev 2.0)[M]. May 1999.
- [8] Han Kedeng, Zhong Liangshu. Verification of AMBA bus model using system verilog[C]. Proceedings of 2007 8th International Conference on Electronic Measurement & Instruments, 2007.1, Xian, China.
- [9] BERGERON J, CERNY E, HUNTER A, et al. Register abstraction layer[M]. Springer, 2005.
- [10] ALEXANDER K, WOLFGANG M. Classification trees for random tests and functional coverage[J]. Design, Automation and Test in Europe, 2006,1:1-2.
- [11] SystemVerilog 3.1a language reference manual. Accellera's Extensions to Verilog®. 2004.
- [12] 黄思远,邵智勇,于承兴,等. VMM中功能覆盖率收敛技术[J]. 集成电路设计, 2010, 33(4):16-17,31.
- [13] 鲁芳,柏娜.基于SystemC和Verilog软硬件协同验证[J]. 现代电子技术, 2008, 31(4):1-3.

(收稿日期:2010-12-21)

作者简介:

章林柯,男,1977年生,博士,主要研究方向:振动与噪声控制及软硬件平台开发研究。

王力,男,1981年生,硕士研究生,主要研究方向:数字集成电路设计与验证、SoC嵌入式系统研究。

王艳武,男,1977年生,博士,主要研究方向:故障诊断研究。