

基于 FPGA 的软硬件协同仿真平台的数据通路设计

徐金娜, 付方发, 王进祥

(哈尔滨工业大学 微电子中心, 黑龙江 哈尔滨 150001)

摘 要: 针对当前 SoC 设计过程中仿真速度过慢的问题, 基于 PLI 机制, 设计了一种能够有效支持基于 FPGA 的软硬件协同仿真平台的数据通路. 其中 PC 端利用仿真工具和 winsock API 构建了激励产生和传输的下行通路, 在 FPGA 端, 利用 Microblaze 组成的 SoC 建立仿真数据加载和结果反馈的上行通路, 同时两端通过以太网实现物理传输. 最后, 上述方案在 Xilinx 开发板实现, 实验结果表明, 该设计能够有效提高仿真效率并且能够支持大规模 SoC 的软硬件协同仿真, 同时具有硬件开销小、通用性强等优点.

关键词: 协同仿真; FPGA; 以太网

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)03-0107-04

Data Path Design of FPGA-based HW/SW Co-simulation Platform

XU Jin-na, FU Fang-fa, WANG Jin-xiang

(Microelectronics Center, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Since the simulation speed is too low for current SoC design process, based on PLI mechanism, we design a data path that efficiently supports FPGA-based HW/SW co-simulation platform. With simulation tools and winsock API, the PC terminal constructs downstream channel that generates and transmits stimulus. With the SoC based on Microblaze, the FPGA terminal constructs upstream channel that is responsible for simulation data loading and results feedback. Both terminals complete physical transmission through Ethernet. Finally, the whole project implemented with Xilinx development board. Experimental results show that this design can effectively improve simulation efficiency and support HW/SW co-simulation of large scale SoC. Meanwhile, the design has the advantages of low hardware cost and strong universality.

Key words: co-simulation; FPGA; Ethernet

1 引言

随着集成电路设计和制造技术的发展, SoC 的功能越来越复杂、规模越来越大, 其仿真验证的难度也随之增加.

软硬件协同验证技术是提高系统仿真速率的一种有效手段. 该技术主要有两种方法: (1) 通过提高仿真所使用硬件模型的抽象程度以实现仿真的简化, 进而达到加速仿真的目的. 虽然该方法能够不同程度的提升仿真速度, 但其代价是换来了仿真精度的降低. (2) 在不考虑设计成本的前提下, 使用 FPGA 等真实硬件来实现 RTL 硬件模型, 它不仅具有

RTL 级的验证精度, 同时它借助 FPGA 能够达到很高运行速度的优势, 使得其仿真速度不会随着设计复杂度增加而有明显的降低.

国内外在进行着软硬件协同验证方面的研究方面取得了一定成果^[1-5], 在提交仿真效率的同时, 增加了系统复杂度.

本文提出了一种基于 FPGA 的软硬件协同仿真平台的数据通路解决方案. 本方案以数据传输效率高的以太网进行双端通信, 使用 EDK 搭建的 SoC 将协同控制模块与待测模块进行的结合, 并且其避免多个 FPGA 的联合使用, 在保证了高效的数据传输的同时简化了设计、节约了硬件开销, 除此之外,

本方案不针对特定模块,具有一般适用性.

2 方案分析

本节将以较常使用的 Modelsim 仿真工具与 FPGA 进行协同仿真为切入点,说明本文所涉及的基于 FPGA 的软硬件协同仿真技术. 仿真过程中,在每个仿真时钟片内对所有门电路输出以及寄存器的值进行计算,每个时钟片内的仿真时间可看作是所有门数的正比例函数,则设表达式为

$$T_M = \alpha N_g \quad (1)$$

式中,参数 T_M 为每个时钟片内的仿真时间, N_g 为系统所用的逻辑门数, α 为 CPU、仿真工具等影响仿真速度的变量的统一表示,为达到不同系统间的一致性与可比性,固视其为定值.

仿真时,整个系统划分为存在于 FPGA 与 Modelsim 中两部分,假设两部分的运算具有绝对的依存关系,且存在于 FPGA 的门资源占总系统的比例为 p ,则

$$p = N_{g(FPGA)} / N_g \quad (2)$$

那么协同仿真的总时间为 FPGA 与 Modelsim 的运算总耗时以及它们之间所涉及的数据传输与控制所需时间的总和,易知:

$$T_{total} = \alpha(1-p)N_g + T_{FPGA} + T_{TRA} \quad (3)$$

式中, T_{FPGA} 为 FPGA 运行所需要的总时间,由于 FPGA 通常可达兆级以上的运算速度,因此该时间不会超过微秒数量级,实际处理时可忽略此时间,故:

$$T_{total} = \alpha(1-p) + T_{TRA} \quad (4)$$

因此得到仿真平台的加速比 ξ (ξ 越小表示平台发挥越大加速作用):

$$\xi = T_{total} / T_M = (1-p) + T_{TRA} / (\alpha N_g) \quad (5)$$

T_{TRA} 是平均每个仿真时间片内数据在 PC 和 FPGA 间所需的传输时间. 同时,仿真一个时钟片的数据会在同一时间传输完毕,所以 T_{TRA} 可以看作是不随系统复杂度改变的固定值.

由式(4)、(5)可以得出以下结论:

(1) 仅减小一个时钟片的 T_{TRA} 就可缩短整个系统的仿真时间. 因此针对一个已固定划分的测试系统,采用以提高软硬件协同验证仿真速率为目的的 FPGA 验证方法的关键在于减小 T_{TRA} .

(2) 仿真环境固定且软硬件划分程度确定时,加速比与系统复杂度有直接关系. 因此,随着系统复杂度的提高,加速比越小, FPGA 方案就越能够发挥加

速作用.

基于以上考虑,本文提出的基于 FPGA 的软硬件协同仿真数据通路的设计利用以太网能够减小 T_{TRA} 以实现高速率数据传输的特点,很好地达到了仿真加速的目的.

3 数据通路的设计

本文提出的软硬件协同仿真平台的解决方案融合了 PLI、以太网、FPGA 等优势能够达到测试加速的目的. 图 1 为该设计的总体结构图,主要由 PC 端和 FPGA 端构成,仿真模块、数据控制模块和协同控制模块是设计的主要部分.

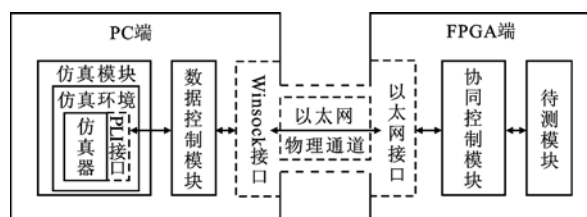


图 1 数据通路结构图

(1) 仿真模块. 仿真模块主要由仿真器和仿真环境组成,二者通过 PLI 进行交互. 该模块主要完成的工作有:①产生待测模块的测试向量以及取得响应;②通过 PLI 完成测试向量和响应在仿真器和仿真环境中的传递;③观察仿真结果.

(2) 数据控制模块. 该模块由 windows 库所提供的网络编程接口函数(winsock API)创建,主要功能有:①获取仿真环境中的测试向量,将其分析后打包发送至以太网;②获得 FPGA 端响应数据,并将其分析后传递给仿真环境;③控制 PC 端的仿真时序,保证仿真的有序进行.

(3) 协同控制模块. 该模块是一个由 EDK 搭建的能够进行以太网传输和数据控制的 SoC,该模块的主要功能有:①通过以太网接收来自 PC 端的测试向量;②将测试向量和时钟正确加载给测试模块;③将响应打包并通过以太网发送给 PC 端;④控制 FPGA 端的数据和时序,确保测试工作的完整性和有序性.

(4) 整体协同工作. 虽然模块间工作是相互独立的,但彼此间有相应的握手信号作为模块间的沟通标识使通路最终能够进行正确的协同工作. 该通路的仿真主要有八步,流程如图 2 所示. 首先,仿真模块将仿真激励传递至数据控制模块,控制模块通过 winsock 接口将激励发送给 FPGA;然后, FPGA 端的协同控制模块将通过以太网取得的激励加载给

测试模块并将相应响应打包发送给PC;最后仿真器将数据控制模块通过winsock接口返回的响应取得,完成一次仿真工作。同时,每进行下一步之前会有一个判断数据是否为当前仿真数据的工作,是则进行下一步,否则等待。

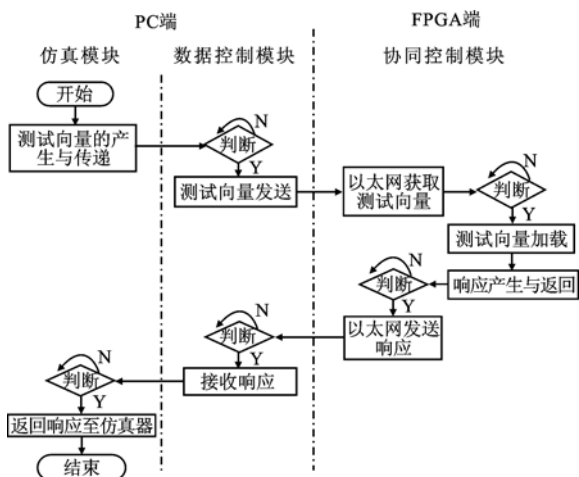


图2 通路的整体协同工作过程

4 具体实现与相关控制

4.1 PC端设计实现

PC端最终完成的工作如图3所示,主要由仿真模块和数据控制模块协同完成。

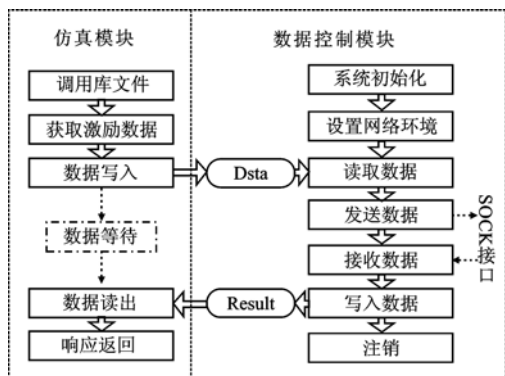


图3 PC端工作过程

(1)仿真模块。由于测试模块移植在FPGA上,因此仅在待测模块的位置例化含有接口和使用PLI子程序编写的用户自定义任务\$moni的空模块。\$moni是实现该模块的重点,它的伪代码如表1所示,其中dat表示激励数据, resu表示当前响应数据。

(2)数据控制模块。主要有缓存区和控制区两部分,缓存区分为测试向量(dat)缓存和响应(resu)缓存,控制区主要完成3节(2)中所述控制工作。此模块中网络环境的相关设置如表2所示,数据控制机

制见4.3节。

表1 \$moni伪代码

```

1:初始化PLI应用:acc_initialize()
2:获得要监视的激励句柄:dat[n],resu[n]
3:while(仿真未结束):
    ①取得激励逻辑值:acc_fetch_value(dat[n], , & dat[n]);
    ②将取得的激励存入缓存区:fwrite(buf_dat, ., dat[])
    ③取得当前响应缓存区响应:fread(buf_resu, ., resu[])
    ④将响应返回仿真器:acc_set_value(resu[], & resu[n], .)
4:关闭PLI应用:acc_close();

```

表2 数据控制模块中的网络设置

	数据缓冲区	通信	协议	收发函数	广播
发送	dat_buf[1024]	IP	UDP	sendto()	true
接收	resu_buf[512]	IP	UDP	recvfrom()	/

4.2 FPGA端设计

协同控制模块是FPGA端设计的主要部分,该模块是一个由EDK搭建的基于MicroBlaze处理器的简易SoC。该SoC采用PLB总线,总线上挂载有MC、GPIO、TIMMER等IP,并集成有用于数据传输的以太网IP和数据控制模块。该模块的具体结构如图4所示。

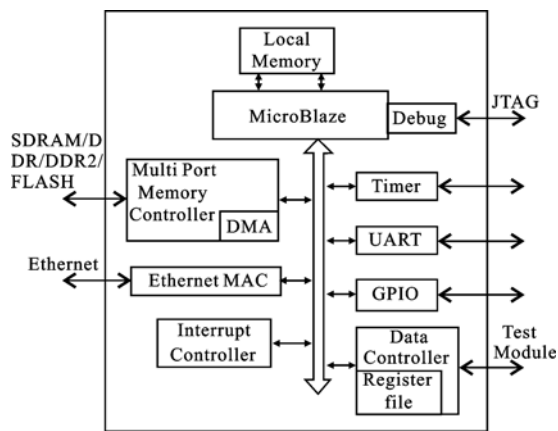


图4 协同控制模块结构

(1)以太网。出于简化设计考虑,本设计中使用IP协议中的不包含较为复杂的流控制和错误控制恢复机制的UDP协议完成以太网数据的传输。传输的数据涉及源IP地址、目的IP地址、源端口号、目的端口号等,这些设置主要由驱动程序完成。根据本设计的需要对其进行相关设置,某次进行传递的以太网帧的主要信息如表3所示。

硬件设备与驱动的结合完成将数据发送到PC端和接收PC端发来数据的工作。发送一个数据到网络上之前,需对该数据进行封装操作。数据封装

时,调用相应的封装函数对发送缓存逐字段进行填充.填充完毕后,将发送缓存中的数据写入发送FIFO,发送FIFO通过PHY层设备将数据发送到网络,并返回发送成功状态.当以太网接收到数据时,首先返回接收成功状态,然后将该数据写入接收缓冲区进行数据解封.数据解封时,调用相应的解封函数对接收缓存逐字段进行检验.若数据通过所有检验,则返回数据段的首地址和数据长度;若数据未通过检验,则将该数据丢弃.对于单次的接收或发送的工作过程如图5所示.虚线框内为数据操作,其他部分主要由驱动完成.

表3 以太网帧信息

目的 MAC 地址	源 MAC 地址	类型	IP 版本
00:24:81:86:43:D1	FF:FF:FF:FF:FF:FF	0x800	0x4
IP 首部长度	长度	协议	IP 首部校验和
0x5	0x23	0x11	0x27C3
源 IP 地址	目的 IP 地址	源端口号	目的端口号
202.118.229.4	169.254.55.65	5150	5150
UDP 长度	UDP 校验和	数据	FCS
0x000F	0x8CBE	0x07010F030E0506	0xD5903241

(2)数据控制模块.该模块分为发送激励和接收响应两个部分.

发送部分主要将新接收的数据加载到测试模块.图6(a)为发送激励模块状态转换图,具体转换过程及条件如下所述:

S0:从S0状态启动,接收到新数据,跳转到S1,否则保持在S0等待.

S1:判断上次数据寄存器中数据是否被发送,是跳转到S2,否则保持在S1等待.

S2:将新数据存入激励寄存器,并跳转到S3.

S3:激励寄存器上的测试向量和仿真时钟加载到测试模块.

S4:等待下一次激励,若没有则返回S0.

接收响应部分的主要功能是从响应寄存器读取响应信息,并将数据进行打包处理,以数据包的形式送到网络中进行传输.图6(b)为接收模块状态转换图.

S0: idle.当有新响应时,跳转到S1状态;否则保持在S0.

S1:将新响应数据存入响应寄存器.

S2:判断响应是否保存完毕,保存完毕则跳转到S3状态;否则保持在S2.

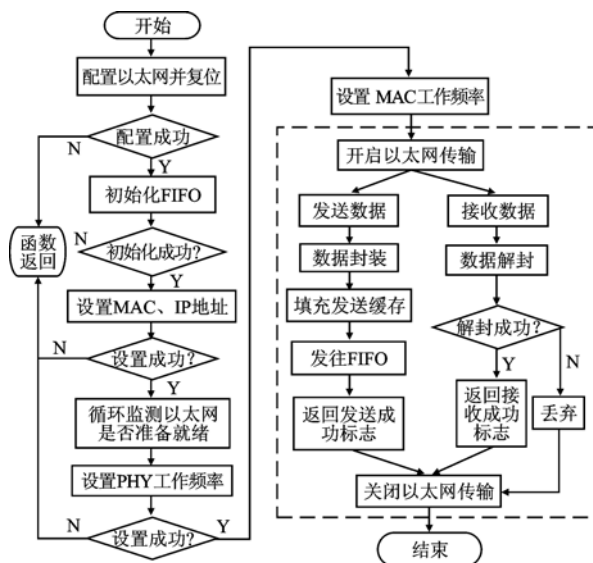


图5 以太网工作过程

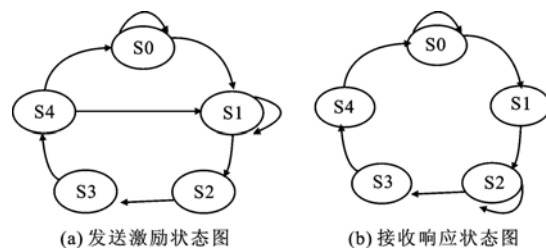


图6 发送与接收状态图

S3: 响应数据接收完毕,取消测试时钟并将响应寄存器中数据打包并通过网络发送.

S4: 等待数据发送状态,并将该状态显示

4.3 数据通路握手机制

在PC端和FPGA端都涉及数据的传递,因此该数据通路能够正确工作的关键在于有序的传递数据.本设计采用握手机制保证这一点,从而避免数据间的竞争.

为了便于讨论,本文将两端的数据收发都归结为图7(a)所示结构,所涉及的时序信息见图7(b).参考图7,当“flag”信号有效时,表示发送端完成了数据的更新,该有效的“flag”信号被接收端采集后,使能“back”信号,通知发送端将更新的数据发送出来,并将数据接收.当整个数据收发过程完成后,清零“flag”、“back”信号,为下一次的数据发送与接收准备.

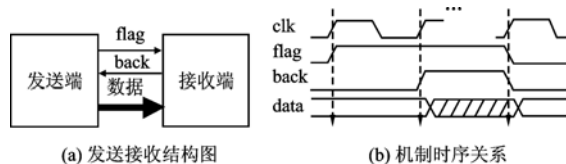


图7 数据通路机制

(下转第114页)

提高. 仿真结果表明:与改进前的算法相比,改进后的三角形质心定位算法具有更高的定位稳定性.

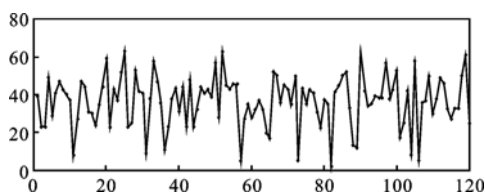


图 5 未知节点定位误差图

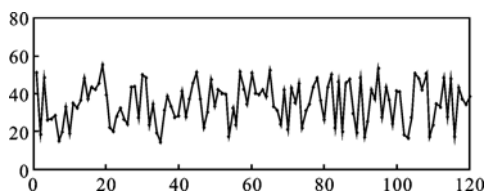


图 6 改进算后的定位误差图

5 结束语

为了解决基于 RSSI 的三角形质心定位算法在减小定位误差上效率较低、定位稳定性不高的问题,本文提出了一种改进的定位算法. 仿真结果表明,在锚节点数目逐渐增加的情况下,改进的三角形质心定位算法在减小定位误差上具有更高的效率和性能,同时也保证了定位的稳定性.

参考文献:

- [1] 刘京. WSN 中质心定位算法和基于移动锚节点的定位算法研究[D]. 郑州:郑州大学,2012.
- [2] 衣晓,刘瑜,邓露. 一种基于 PIT 的无线传感器网络质心定位算法[J]. 传感技术学报,2010,23(7):1012-1016.
- [3] 孙懋珩,廖根健. WSNs 中基于粒子群优化的节点定位算法[J]. 测控技术,2011,30(12):111-115.
- [4] 冯爱丽,乔钢柱,曾建潮. 基于信标节点间距离的改进 RSSI 定位算法[J]. 太原科技大学学报,2012,33(1):6-9.
- [5] Ming YAN, Hao ZHU, Liangquan JIAO, et al. An optimized algorithm of triangle centroid localization based on RSSI ranging[C]//2011 International Conference on Information, Services and Management Engineering [s. l.]:IEEE,2011:1458-1461.
- [6] 林玮,陈传峰. 基于 RSSI 的无线传感器网络三角形质心定位算法[J]. 现代电子技术,2009,289(2):180-182.
- [7] 邓彬伟,黄光明. WSN 中的质心定位算法研究[J]. 计算机应用与软件,2010,27(1):213-214.

作者简介:

郭 瑞 女,(1974—),博士生,副教授. 研究方向为电气传动系统自动化技术、现代调速理论及应用.

马寅飞 男,(1986—),硕士研究生. 研究方向为电力电子、过程控制.

(上接第 110 页)

5 实验

为验证该设计的可行性和正确性,本文采用加法器作为待测模块并将其连同协同控制模块放置 Xilinx 开发板上进行协同仿真. 从实验结果看出整个通路完成了正确的加和作用. 虽然仅用加法器作为待测模块,但足以说明该通路的可行性,同时加法器为非特定设计,证明了该设计的通用性.

6 结束语

本文设计了一种软硬件协同仿真平台数据通路,该通路继承了 PLI、以太网、FPGA 等的优越性,不仅能够对仿真起到加速作用,更有硬件消耗低、通用性强等优点. 使用实例证明了该通路的可用性. 该方法的提出对于完善整个仿真加速平台具有重要意义.

参考文献:

- [1] Joseph B, Soonhoi H, Edward A L, et al. Ptolemy: a mixed-paradigm simulation/prototyping platform[C]//

Proceedings of the C++ At Work Conference. [s. l.]: IEEE,1991:1-16.

- [2] 冯博凌. 基于混合建模方法的 SoC 软硬件协同验证环境的设计研究[D]. 成都:电子科技大学,2009.
- [3] David A. Research accelerator for multiple processors (RAMP)-a shared experimental parallel HW/SW platform [C]// IEEE International Symposium on Performance Analysis of Systems and Software. [s. l.]:IEEE,2006:1-20.
- [4] Chung E, Nurvitadhi E, Hoe J, et al. ProtoFlex-FPGA-Accelerated Hybrid Functional Simulator [C]// Computer Architecture Lab at Carnegie Mellon Technical report. Long Beach, CA:IEEE,2007:1-6.
- [5] 汤江逊. SoC 软硬件联合仿真验证系统研究[D]. 成都:电子科技大学,2006.

作者简介:

徐金娜 女,(1986—),研究生. 研究方向为成系统与 SOC.

付方发 男,(1984—),博士,讲师. 研究方向为集成系统与 SOC.

王进祥 男,(1968—),博士,教授. 研究方向为集成系统与 SOC.