《电子技术应用》 www.ChinaAET.com

Perl 在 ESD 保护电路中研究与应用

蒋小平，阳 鹏
(中国矿业大学 机电与信息工程学院，北京 100083)

摘 要：研究了 Perl 在 ESD 保护电路中的应用。基于 Perl 语言的强大功能，在海量的数字电路仿真数据中准确地抓取需要的数据，并生成文件报表。同时为数字仿真电路的验证提供了一种全新、快速、准确的方式。

关键词：Perl；ESD 电路；电路验证
中图分类号：TN43 **文献标识码**：A **文章编号**：1674-7720(2014)04-0023-03

The research and application of Perl language in ESD protection circuits

Jiang Xiaoping, Yang Peng
(School of Mechanical Electronic & Information Engineering, China University of Mining and Technology, Beijing 100083, China)

Abstract: This paper studies the application of Perl in the ESD protection circuit. Based on the powerful function of Perl language, the needs data is grasped accurately in the massive data file of the circuit, and the report is generated. And it provides a new, fast and accurate authentication for digital simulation circuit verification.

Key words: Perl; ESD protection circuit; circuit verification

在一款电子产品投入生产前，数字后端设计工程师会对数字仿真电路进行验证，对整个版图的线宽、间距、短路等情况进行检查，从而确保生产的产品都能够正常工作。这些仿真电路验证工作经常需要面对海量的电路数据进行筛选，识别出设计参数不合格的电路并进行改善[1]。

数字仿真电路的设计离不开仿真验证，仿真验证过程是一个不断重复的过程，它贯穿了整个仿真电路的设计。据统计，设计验证会占用设计人员 70%以上的精力，而且对于大型的电路，验证会更为繁琐，例如门级规模上亿的电路。

1 数字仿真电路验证方法分析

作为可编程逻辑器件，FPGA(Field Programmable Gate Array)相比于 ASIC(Application Specific Integrated Circuit)可以重复擦写程序，通常用它来对现有的设计进行仿真验证。在数字 ASIC 的设计中，由于测试向量的不完备，设计中很有可能存在相应的 Bug。因为每次 ASIC 的投片会花费较大代价，为规避风险，通常会对设计进行 FPGA 验证，FPGA 验证无误后才进行投片、量产[2]。

目前 FPGA 数字后端验证的方法主要是基于人工的选择，通过 Linux 下的条件查找功能，逐个实现对电路参数的验证功能。要验证的电路数目和需要验证的电路参数个数都是海量的，而且需要验证的参数类型也是随时变化的，因此这种人工验证的方法，不仅效率低下，而且验证准确度难以确保。

本文将介绍一种新的利用 Perl (Practical Extraction and Reporting Language)进行数字仿真电路验证的方式。

2 Perl 介绍及其在 ESD 保护电路中的应用

2.1 Perl 简介

Perl 是解释型的脚本语言，它具有出色的处理文本能力，是 Windows 和 Linux 跨平台的编程语言，具备自主内存管理功能，没有内存泄漏问题，且具有强大便捷的模块化功能[3]。

Perl 的解释程序是开放源码的免费软件，使用 Perl 不必担心费用；Perl 能在绝大多数操作系统下运行，可以方便地向不同操作系统迁移；它可以作为用在不同系统环境编程的高级语言，为电路验证提供了一种新的解决方式；其高效、准确的特点给设计人员带来了很大的便利[4]。

2.2 ESD 保护电路

静电放电 ESD(Electrostatic Discharge)会给电子器件环境带来破坏性的后果，是造成集成电路失效的主要原因之一。

在正常工作情况下，NMOS 横向晶体管不会导通。当

《电子技术应用》 www.ChinaAET.com

ESD 发生时，漏极和衬底的耗尽区将发生雪崩，并伴随之产生电子空穴对。一部分产生的空穴被源极吸收，其余的流过衬底。由于衬底电阻 R_{sub} 的存在，使衬底电压提高。当衬底与源之间的 PN 结正偏时，电子就从源发射进入衬底。这些电子在源漏之间电场的作用下被加速，产生电子、空穴的碰撞电离，从而形成更多的电子空穴对，使流过 n-p-n 晶体管的电流不断增加，最终使 NMOS 晶体管发生二次击穿，此时的击穿不再可逆，则 NMOS 管损坏[5]。

为了进一步降低输出驱动上 NMOS 在 ESD 时两端的电压，可在 ESD 保护器件与 GGNMOS 之间加一个电阻。这个电阻不能影响工作信号，因此不能太大。画版图时通常采用多晶硅(poly)电阻 $R1$ 和 $R2$。如图 1 所示。

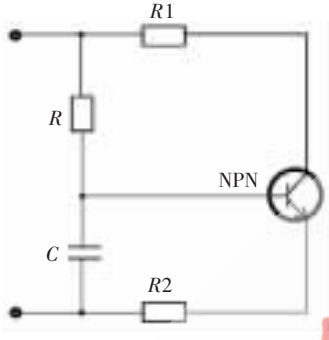

图 1 ESD 保护电路的结构

2.3 Perl 在 ESD 保护电路验证中的应用

Perl 在 ESD 保护电路验证中可以实现抓取所需要的数据信息并生成文件报表。图 2 所示为一款芯片的 ESD 保护电路报表中的一个小模块仿真电路，其整个报告有上千个格式相同而 $R1$、$R2$ 数字不同的电路小模块。由图 1 所示的电路说明，要实现对芯片的 ESD 保护，需要限定：$R1$ 或者 $R2$ 不能大于某一个限定值 R_m。因此需要在这个文件报表中实现不满足此条件（即：$R1$ 或 $R2$ 电阻值有一个大于 3.5 Ω）的所有电路 Location、Power 和 Ground 的信息抓取。如果不用脚本程序，只能对电路模块的 $R1$ 和 $R2$ 数值信息一个一个分析比较，这样不仅准确率低，容易出错，而且耗费大量后端工程师的精力。但是现在可以通过一个 Perl 程序实现对此类

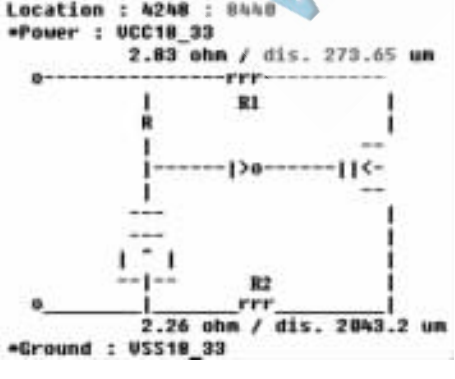

图 2 ESD 电路信息报告

文件报告进行条件抓取。这种方式不仅运行速度快，不会出错，而且大大减少了后端验证工程师的数据分析工作量。

3 Perl 的 ESD 保护电路验证实现

经以上分析，采用 Perl 语言编写的处理子程序如下[6]：

```
#! /usr/bin/perl
….
sub process()
{
print "please input a file_name:\n";
$file_name = <STDIN>;
open(IN,"$file_name")||die("can not open the file! $! n");
my $nl=$/;$/=$/.$/;
while(my $line=<IN>)
{my($ohm1,$ohm2,$location,$ground_name,$power_name);
my @line = split/$nl/,$line;
$line[2] =~/.*Location : (.*) : (.*)/ms;
$location= join(":",$1,$2);
$line[3] =~/.*Power : (.*)/ms;
$power_name =$1;
$line[4] =~/^s{0,}([\d\.]+)/;
$ohm1=$1;
$line[16] =~/^s{0,}([\d\.]+)/;
$ohm2=$1;
$line[17] =~/.*Ground : (.*)/ms;
$ground_name =$1;
my($power,$ground);
if($ohm1>$level||$ohm2>$level){$power=$ohm1;
$ground=$ohm2;
…
close (IN);
```

运行这个 Perl 脚本后，只需要输入电源电阻所需要满足的条件，不同系列的芯片对 $R1$ 和 $R2$ 要求不一样。譬如，对于某一系列芯片需要查找出 $R1$ 或者 $R2$ 大于 3.5 Ω 的所有电路 location、power、ground 信息(注：对于此系列芯片 $R1$ 或者 $R2$ 大于 3.5 Ω 时的 ESD 保护电路都是不能通过检查的，需要重新设计参数)。Perl 可以在 Windows、Unix 等多种平台下运行，以在 Windows 系统下运行为例[7]，如图 3 所示。

图 3 Windows 下运行 Perl 脚本

最后的运行 Perl 脚本输出为一个 result 文件名的文件报表，结果如图 4 所示。

图 4 文件报表

数字仿真电路验证是其数字后端设计中极其重要的一部分，是保证每款芯片品质的重要部分。Perl 语言可以很好地实现所有不满足 ESD 保护电路设计的电路参数抓取，方便数字仿真电路设计后端查错。由此表明可以根据不同电路的实际仿真报告，用 Perl 编程实现不同数字仿真电路验证的后端查错，大大减少了验证者的工作量。可以预见，以后的硬件电路设计中会越来越多地见到 Perl 的身影[8]。

参考文献

[1] 余晓文，强英.Perl 语言在电路设计中的应用[J]. 微型机与应用，2004，23(1):24-25.

[2] 许川佩，唐海，胡聪.基于 FPGA 的 NoC 硬件系统设计[J]. 电子技术应用，2012，38(2):117-119.

[3] (美)施瓦茨. Perl 语言入门(第 6 版)[M].盛春，译.南京：东南大学出版社，2012.

[4] (美)艾德尔曼.使用 Perl 实现系统管理自动化(第 2 版)[M]. 盛春，译.南京：东南大学出版社，2011.

[5] 孙可平，刘勇.我国电子行业 ESD 防治技术现状及对策[J]. 上海海运学院学报，1999(1):76-81.

[6] DEITEL H M. Perl 编程金典[M]. 李晋宏，杨小平，译. 北京：清华大学出版社，2002.

[7] 华胜华，刘伟平. PERL 在 IC 设计中的应用[J].中国集成电路，2004(5):36-41.

[8] WALL L, CHRISTIANSEN T, ORWANT J. Programming Perl[M]. Third Edition, O'Reilly Media, 2000.

(收稿日期：2013-11-29)

作者简介：

蒋小平，男，1966 年生，副教授，硕士生导师，主要研究方向：控制理论与控制工程。

阳鹏，男，1991 年生，硕士，主要研究方向：控制理论与控制工程。

APPLICATION OF ELECTRONIC TECHNIQUE www.ChinaAET.com