

基于HDL的32位ALU设计实例的数字电路测试

赵泽卿

北京海淀尚丽外国语学校

DOI:10.32629/acair.v4i1.19377

[摘要] 由于电子技术发展迅速,现在电子行业和学术界都开始利用硬件描述语言(HDL)进行仿真测试。本文首先介绍数字电路测试的基本原理,随后以一个32位算术逻辑单元(ALU)芯片为研究对象,搭建一个虚拟仿真测试平台。在这个平台里测试10轮数据,发现芯片错误数量从最初的50个显著下降至20个以下,同时测试覆盖率稳步提升。成功验证了ALU的加减乘除及逻辑运算功能。

[关键词] 数字电路测试; 硬件描述语言(HDL); ALU; 仿真验证; 故障检测

中图分类号: U226.5+1 **文献标识码:** A

Digital Circuit Testing of a 32-bit ALU Design Example Based on HDL

Zeqing Zhao

BEIJING HAIDIAN SHANGLI FOREIGN LANGUAGE SCHOOL

[Abstract] Due to the rapid development of electronic technology, both the electronic industry and the academic community have begun to use hardware description languages (HDL) for simulation testing. This paper first introduces the basic principles of digital circuit testing, and then takes a 32-bit arithmetic logic unit (ALU) chip as the research object to build a virtual simulation testing platform. In this platform, 10 rounds of data were tested, and it was found that the number of chip errors decreased significantly from the initial 50 to less than 20, while the test coverage steadily increased. The addition, subtraction, multiplication, division, and logical operation functions of the ALU were successfully verified.

[Key words] Digital circuit testing; Hardware description language (HDL); ALU; Simulation verification; Fault detection

1 引言

从飞驰的高铁、智能汽车,到复杂的通信基站,这些设备内部都隐藏着一种专门的“微型电脑”,我们称之为嵌入式系统^[1]。它的核心任务就是对实时的事件进行监控、响应和控制。这是一种“软件+硬件”的组合物。随着硬件技术的飞速进步,现在我们已经可以在电路中嵌入这种微型计算机,并像写软件一样去“编程”硬件。为了构建这些复杂的嵌入式系统,工程师需要设计各种数字逻辑电路,如专用集成电路(ASIC)、微处理器和可编程逻辑器件(PLD)。硬件描述语言正是用于描述这些数字电路结构和行为的工具。它让设计者能够用代码的形式来定义电路的功能。电路设计完成后,必须进行测试。这么做的目的是检查系统的某一部分或模型是否偏离了设计规格说明^[2-4]。如果逻辑电路不按照标称行为运行,就说明该电路失效^[5]。测试数字电路的目的是发现故障,并确保电路按预期工作。

得益于超大规模集成电路(VLSI)技术的进步,我们现在已经能够在一片单芯片上制造出极其复杂的数字逻辑电路^[6]。但是,开发这样的系统需要考虑很多问题。一个数字系统的诞生,

始于使用HDL进行设计,终于最终的制造与发货。在这个漫长的过程中,工程师必须持续执行仿真、综合、测试及维护工作,不仅要确保功能正确,还要兼顾性能等非功能方面的要求。

数字电路可以设计为逻辑元件的互连。逻辑元件的一些例子包括与门、或门、寄存器、反相器、触发器等^[7]。数字电路必须能够处理有限值的离散信号。数字电路与模拟电路不同,因为后者由晶体管、电阻器和电容器等电子元件组成,处理具有连续值的信号,如电流和电压。模拟电路的一个例子是运算放大器,而二进制计数器则是数字电路的一个例子。

Wang^[8]将数字电路的测试定义为检测系统中的故障并定位这些故障以便修复的过程。通常,针对一个被测电路,会被称为测试向量或测试模式的测试激励施加到其输入端。通过收集输出响应并将其与预期响应进行比较来分析。如果实际行为偏离了预期行为,则表明检测到了故障。

2 测试方法设计

2.1 准备测试环境

搭建仿真平台,需要安装好仿真软件。之后把设计好的电路

代码和用来检测它的测试代码导入进去,并设置好时钟信号和时间精度,确保电路能按预定的节奏工作。最后,在正式测试前要先进行一次“试运行”,只要能跑通最简单的指令并顺利画出波形图,就说明这个测试平台搭建成功了。

2.2 配置测试软件

配置软件需要先设定时钟频率和持续时间。接着,开启随机测试生成器,使得程序可以根据我们设定的范围约束条件自动产生各种各样的输入数据,再设置一些故障注入,来考验芯片的应变能力。最后,打开覆盖率统计,这样仿真结束时就能知道测试是否覆盖了芯片的所有功能。

2.3 确定测试顺序

执行测试按顺序分为启动、输入、监控和记录四个步骤。在启动步骤中,需要向仿真软件中加载电路设计文件和测试代码,并确认电路已正确复位到初始状态。接着,我们将预设的测试数据输入电路,有时甚至会故意模拟一些故障信号来进行检测。一旦捕捉到错误便会自动记录。最后,仿真结束时,系统会将所有输出结果和统计数据打包保存为日志,以便我们后续分析电路是否合格。

2.4 芯片性能分析

仿真结束后,我们需要对芯片进行四个维度的评估。首先是功能检查,将芯片的实际输出与参考模型逐一比对,确认计算结果完全正确,不存在逻辑错误。其次是覆盖率分析,统计测试进程是否覆盖了所有的代码行和功能点。接着是性能与时序分析,检查芯片的运算速度、反应时间是否达标,以及信号传输是否稳定。最后是可测试性评估,判断这次测试所花费的时间和资源是否合理,以此判定芯片是否可以通过验收,还是需要优化设计。

2.5 测试流程

本节将基于上一章提出的测试流程框架,对某型32位算术逻辑单元芯片MYC-ALU32进行测试流程设计与论证。通过实例化流程步骤,展示该流程在实际HDL电路验证中的具体操作方式与评价方法。

2.6 硬件与软件环境

为了验证32位算术逻辑单元(ALU)芯片的功能,我们在计算机上搭建了仿真实验环境。具体的软硬件配置如表1所示。

表1 实验环境配置表

环境类型	配置
仿真平台	ModelSim/QuestaSim 2024 版
HDL 类型	Verilog + SystemVerilog Testbench
目标模块	myc_alu32.v
支持文件	操作码定义、参考黄金模型、激励生成脚本
系统资源	Win/Linux 主机,8GB+ 内存用于长时间仿真

本次测试的目标芯片MYC-ALU32是一个计算核心,其主要信号端口如下:

输入端: 包含两个32位的数据输入(A和B),一个5位的操作指令输入(OP),以及时钟(CLK)和复位(RST)信号。

输出端: 包含32位的计算结果(OUT),以及三个状态标志位: 零标志(FLAGS_Z)、进位标志(FLAGS_C)和溢出标志(FLAGS_V)。

Inputs: A[31:0], B[31:0], OP[4:0], CLK, RST

Output: OUT[31:0], FLAG_Z, FLAG_C, FLAG_V

我们在ModelSim软件中将时间精度设置为1纳秒(1ns/1ps),每组测试的运行时间设定为1000纳秒。仿真过程中,软件会自动记录所有信号的变化,并生成波形文件(WLF)供后续查看。

为了全面测试芯片,我们利用SystemVerilog语言的“随机化功能”来自动生成测试数据。如下列代码所示,我们定义数据A、B和指令OP为随机变量,但通过constraint(约束)语句,限制OP的取值必须在0到11之间,以确保生成的指令都是芯片能识别的有效操作。

```
rand bit[31:0] A, B;
rand bit[4:0] OP;
constraint op_valid {
  OP inside {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
}
```

2.7 确定测试顺序

第一步是基础复位检查。我们先给ALU发送复位信号,确认输出端归零。随后输入全0数据(OP=0, A=0, B=0),观察芯片输出是否保持为0。第二步是功能验证。针对芯片支持的各类运算,设计不同的指令编码配合典型数据组合(测试向量),具体方案见表2。

表2 功能测试规划表

OP 编码	功能	典型测试向量
0	ADD	1+1, 最大值+1, 溢出测试
1	SUB	2-1, 0-1, 借位测试
100	AND	边界位全 1/0 组合
110	SHIFT L	最高位变化测试

对于上面提到的每种运算,我们都准备了四种类型的题目: 边缘测试用于测最大值、最小值等极端数据。随机测试由电脑随机出题,模拟各种杂乱的情况。负数测试专门检查负数运算是否出错。边界测试检查数据在寄存器边缘时的表现。

准备就绪后,我们在软件中输入命令vsim -c myc_alu32_tb -do "run-all"启动全自动仿真。在测试运行期间,记录所有的波形数据(WLF文件)、错误报警日志以及测试覆盖率数据。

2.8 结果与分析

通过观察图1,我们可以看到芯片优化过程总体趋势一路下降。柱状图的高度随着测试轮次逐渐变低,这说明芯片系统正逐步走向稳定。ALU与IO端口错误占比最高。因为ALU逻辑最复杂,负责繁重数学运算。IO端口时序要求严苛,故障极易发生。时钟和存储模块部分错误很少,这类模块多采用业界成熟标准设计,自身可靠,出错概率低。

图2展示了随着测试轮次的增加,芯片覆盖率是如何逐步提升的。FSM代表“状态机覆盖率”,它检查芯片几个大状态之间

是否切换正常。因为状态数量很少,所以简单易测。Line和Toggle对应代码行及信号翻转情况。随着测试时间拉长,覆盖范围扩大,所以这两条曲线是稳步上升的。Branch一直处于最低的位置。这是因为“分支覆盖率”检查针对复杂逻辑判断。随机测试难以命中极端条件组合,得分率通常是最低的。

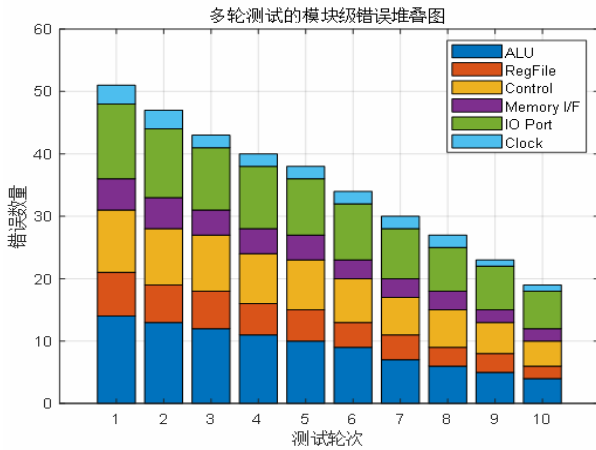


图1 模块级错误数量趋势

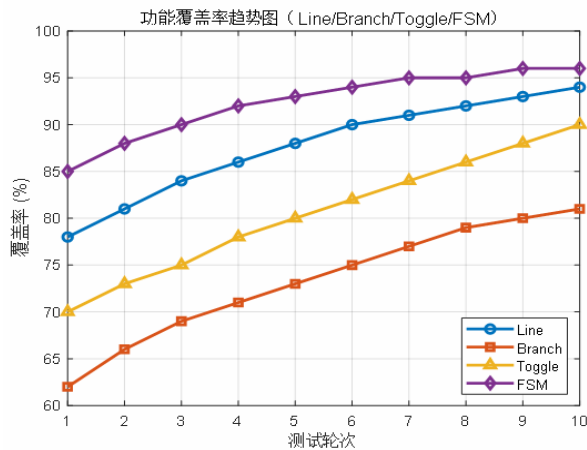


图2 覆盖率趋势图 (Line/Branch/Toggle/FSM) 分析

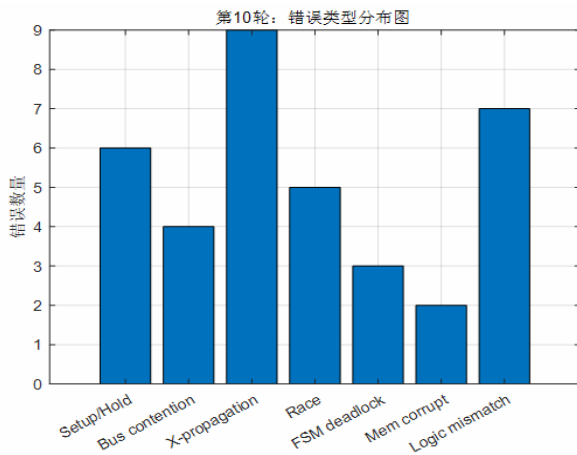


图3 第10轮错误类型柱状图分析

由图3可以看出,第10轮的错误类型主要有以下几类问题:未知状态扩散是目前占比最高的一类错误。在数字电路中,如果一

个信号既不是0也不是1,我们就叫它“X”(未知态)。这通常是因为有些寄存器在刚通电时没有被正确复位,这种状态会在电路里传染,导致后续结果全部出错。

逻辑计算错误排在第二位。这说明在某些复杂的运算逻辑中,芯片代码本身还存在漏洞,需要继续修改逻辑。时序违例反映的是时间问题。芯片内部的信号传递需要极高的时间精度。如果信号跑得太慢,就会导致数据读取失败。我们同时也发现,状态机死锁和内存损坏的错误非常少。这证明芯片的基础架构是非常稳固的。

3 总结

本文研究了硬件描述语言(HDL)的数字电路测试生成技术,并通过一个32位运算芯片(MYC-ALU32)的实际测试案例验证了这套方法的有效性。实验结果表明,通过搭建虚拟仿真平台并运用随机测试和故障注入技术,我们能够有效地发现电路中的潜在缺陷。本文提出的测试流程能够帮助工程师在芯片排除故障提供参考。

[参考文献]

[1]Fan Y H. Design and Implementation of Test System for System Chip Function Test[C].2023 3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC),2023.

[2]Rao S R,Pan B,Armstrong J R.Hierarchical test generation for VHDL behavioral models[C]. European Conference on Design Automation,1993:175-182.

[3]Riahi P A,Navabi Z,Lombardi F.A VPI-based combination al IP core module-based mixed level serial fault simulation and test generation methodology[C].IEEE Test Symposium,2003: 274-277.

[4]Lynch M L,Singer S M.A next generation diagnostic ATPG system using the Verilog HDL[C]. IEEE International Verilog HDL Conference,1997:56-63.

[5]Bernardon D P.Real-time reconfiguration of distribution network with distributed generation[J].Electric Power Systems Research,2014,107:59-67.

[6]Lin C C.Energy-efficient task scheduling for multi-core platforms with per-core DVFS[J]. Journal of Parallel and Distributed Computing,2015,86:71-81.

[7]Huang W K,Lombardi F.An approach for testing programmable/configurable field programmable gate arrays[C].14th VLSI Test Symposium,1996.

[8]Kalomiros J A, Lygouras J. Design and evaluation of a hardware/software FPGA-based system for fast image processing[J].Microprocessors and Microsystems,2008,32(2):95-106.

作者简介:

赵泽卿(2006--),男,汉族,河北保定人,本科,研究方向:英语教育。