

基于 ARINC629 的层次化验证平台设计

王 鹏^{1,2}, 邵 伟³, 姜承翔³, 田 毅¹, 薛茜男¹

(1. 中国民航大学 天津市民用航空器适航与维修重点实验室, 天津 300300;

2. 民用飞机模拟飞行国家重点实验室, 上海 200232; 3. 中国民航大学 安全科学与工程学院, 天津 300300)

摘要: 结合约束随机、覆盖率驱动及断言等多种验证方法, 对层次化验证平台的搭建方法进行了研究; 以对 ARINC629 航空数据总线接口控制器的验证过程为例, 介绍了基于 System Verilog 的层次化验证平台的搭建过程及其使用方法; 根据验证后覆盖率报告, 证明各项功能均已达到其设计需求, 且实现率为 100%; 验证结果表明该平台和传统验证方法相比极大地提高了验证效率, 平台的搭建方法和结构具有一定的通用性, 更改其数据生成模块即可用于其他类似设计的验证。

关键词: SystemVerilog 语言; 功能覆盖率; 断言; ARINC629 航空数据总线

Design of Hierarchy Testbench Based on ARINC629

Wang Peng^{1,2}, Shao Wei³, Jiang Chengxiang³, Tian Yi¹, Xue Qiannan¹

(1. Tianjin Key Laboratory for Civil Airworthiness and Maintenance, Tianjin 300300, China;

2. State Key Laboratory for Civil Aircraft Simulated Flight, Shanghai 200232, China;

3. College of Safety Science & Engineering, Civil Aviation University of China, Tianjin 300300, China)

Abstract: A method to build a Hierarchy Testbench is presented in this paper, which uses constraint-random, coverage-driven, assertion-based methodology. According to the coverage report after verification, it proved all the functions have been reached perfectly. The verification of ARINC629 interface controller proved it to be more efficient, with explaining the design and application of this testbench, better than the traditional verification method. The methodology and the structure can be used in other similar design, with only changing its generation module.

Key words: systemVerilog; coverage; assertion; ARINC629

0 引言

随着集成电路制造工艺技术的发展, 定向验证越来越难满足现代复杂电路发展的要求。为节省验证时间和人力, 需要尽早地找出设计中的逻辑错误。据统计, 对于一款复杂的系统设计, 设计验证时间占整个开发周期的 70% 左右^[1], 因此如何在流片之前搭建一个正确完备及耗时短的验证平台成为了设计成功的关键。传统的验证方法要求验证工程师编写大量的定向激励, 通过人工的方法检查仿真结果, 这种验证方法不仅效率低下, 难以达到满意的覆盖率, 而且针对不同的设计模块需要重新设计验证平台, 可重用性差。

ARINC629 航空数据总线作为新型航空总线克服了 ARINC429 传输速率慢的缺点, 又没有采用 1553B 的集中式控制, 在波音 777 上得到成功地应用^[2]。本文采用基于 SystemVerilog 语言的分层次验证方法, 完成针对 ARINC629 接口控制器的验证平台搭建。通过分析验证结果, 此验证平台能够对 ARINC629 控制器进行全面且有效的验证, 并具有一定的重用性。

1 ARINC629 接口控制器

ARINC629 规范定义了采用避免冲突的载波监听多路存取

(CSMA/CA) 总线访问方式, 其终端发送模块可以对来自处理器的并行二进制数据进行编码, 加上同步字头和奇偶校验位后发至总线。当接收逻辑模块检测到有效的同步字头时, 接收该信息帧并对其译码, 同时进行串/并转换及奇偶校验等, 其中, 整个 ARINC629 终端的技术核心就是其编码和解码功能。

ARINC629 接口控制器是指完成总线数据接收和发送功能的控制器, 包括接收模块、发送模块和协议模块^[3]。控制器连接包括全局端和局部端, 分别与其他控制器和用户通信, 如图 1 所示。



图 1 ARINC629 控制器

ARINC629 控制器具有编码、解码、接收、发送、故障诊断及码型转换等功能。ARINC629 总线采用曼彻斯特双向电平编码方式, 即编码 1/0 代表逻辑 1, 编码 0/1 代表逻辑 0^[4]。以解码过程为例, 整个过程由状态机控制, 其逻辑框图如图 2 所示, 编码过程为解码的逆过程, 其原理相似。

图 2 中, Seridata 为串行输入数据, 当在跳变后检测到大于一个周期的高(低)电平时, 认为同步字头有效, 则同步头检测位 Syntest 置低电平, 同时进行时钟分离、码型转换及移位。当移位完成后进行奇偶校验, 同时把数据存入保持寄存器, 然后将数据准备位 Dataridy 置高电平有效, 表示解码完成。此时, 接收模块将发送 Dataridy 信号并进入下一个解码周期, 等待新的数据输入。

收稿日期: 2013-10-21; 修回日期: 2014-01-09。

基金项目: 国家自然科学基金委员会与中国民用航空局联合资助项目(U1333120); 中央高校基本科研业务费项目(312013SY53); 中国民航大学科研启动基金项目(2012QD26X)。

作者简介: 王 鹏(1982-), 男, 副研究员, 主要从事民机系统安全性设计与评估、机载电子硬件适航技术方向的研究。

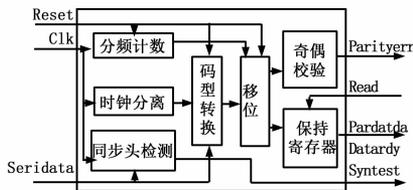


图 2 解码逻辑框图

2 验证平台

传统验证平台只有两个机制，即发送激励和响应检查，所以其传统验证平台主要由两个部分组成：驱动器和监视器。其结构简单，可重用性差，功能过于集中，不仅使得开发任务变得繁重，更加大了验证的难度^[6]。

为提高验证效率，本平台采用基于 SystemVerilog 语言的分层次验证架构^[6]。SystemVerilog 已成为业界硬件描述和验证的标准语言，可支持约束随机测试、覆盖率统计分析、断言验证等，有效的提高了验证的可靠性、完整性及可重用性。

2.1 分层次验证平台

本平台的搭建如图 3 所示，主要包括生成器 (Generator)、驱动器 (Driver)、监视器 (Monitor)、记分板 (Scoreboard) 和断言 (SystemVerilog Assertion, SVA) 组成。和传统的验证平台相比，本平台实现了验证平台的高层次、抽象化发展，使得平台的各个功能分化开来并通过类 (class) 封装在特定的模块中。这样一来就使得验证环境独立于待测对象，在验证环境的上层生成测试案例，实现了验证平台的可重用性。图 3 为验证平台结构图，对于不同的待测设计，无需修改环境只改变测试案例就用于验证。较之于传统验证平台，这样不仅减少了验证代码的编写数量，同时也降低了验证人员的工作量和出错的概率。

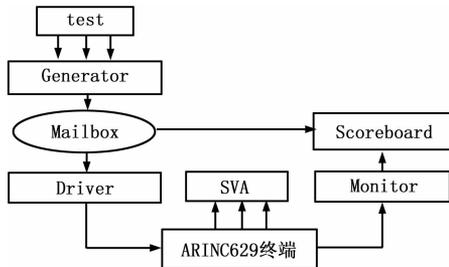


图 3 分层的验证平台结构

本平台搭建过程中，除断言 (SystemVerilog Assertion, SVA) 和邮箱 (Mailbox) 外，环境 (Environment) 内的其他组件都以类的形式实现。在顶层通过接口模块实现验证平台与待测设计 (DUT) 之间的通信，其他事务的执行都被封装在验证环境中。在验证环境中需依次实现下列功能：

- (1) 对接口实例化；
- (2) 根据协议初始化验证环境；
- (3) 对验证环境内的其他模块实例化；
- (4) 执行验证任务；
- (5) 收集运行信息并生成报告。

2.2 线程同步

图 3 中邮箱并不是一个模块，从硬件角度讲，它相当于具有源端和收端的 FIFO^[1]。在本平台中多处使用了邮箱，并利

用握手事件来实现两个线程间的同步，保证 ARINC629 控制器和平台间数据协同步调，按预定的先后次序运行。

以对 ARINC629 控制器发送模块输入信号为例，为保证当前生成的数据即为控制器接收到的信号，在生成器和驱动器间使用邮箱 gen2drv 通过握手事件 drv2gen 来实现，只有驱动器对 ARINC629 控制器完成一组信号驱动之后，才允许数据生成器生成下一组数据，这样做有利于保证每一组相应都能找到对应的激励，在需要的时候又有利于加入故障注入。其工作原理如图 4 所示。

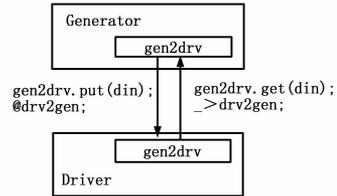


图 4 邮箱同步示意图

2.3 约束随机测试

本平台采用约束随机测试，需要通过邮箱将对期望值传到记分板 (Scoreboard) 与经过 ARINC629 控制器的实际输出数据作比较，整个过程的执行顺序如下：

- (1) 约束随机产生自动测试集；
- (2) 将测试数据送到总线接口；
- (3) 实时监视总线输出数据并将数据传送给记分板；
- (4) 保存总线的输出数据；
- (5) 将不同测试数据的输出分别和对应的期望值比较。

在传统的验证平台中，测试案例一般是由验证人员根据设计需求逐条编写，这种定向测试集可以找到功能项中可能存在的问题，但是当功能项的数目很多时，这种方法将很难按清单检查功能项的方法检查功能项潜在的问题^[7]。本平台的解决方法是采用受约束的随机测试法产生自动测试集，并通过随机激励找到潜在的问题。通过在随机激励中加入约束条件，可以提高测试向量的有效性和准确性^[8]。

以 ARINC629 控制器发送模块输入信号同步头选择位 cmd 为例，高电平表示产生标号字同步头，低电平表示产生数据字同步头。在实际验证过程中，为了提高验证效率，其中要求：同步头选择位 cmd=1 的概率是 cmd=0 的 10 倍，其添加约束代码如下。

```
rand logic cmd;
constraint d1{cmd dist {0 := 1, 1 := 10};}
```

2.4 SVA 断言

SVA 语言 (SystemVerilog Assertion) 是一种描述性语言，可以完美地描述时序相关的状况。SVA 本身简洁易读，便于维护，而且还提供了许多内嵌的函数用于测试特定的时序关系和自动收集功能覆盖率的数据，往往只需要短短几行代码就可以检查同一类时序问题。在本平台中，断言以 module 的形式单独实现，独立于环境类之外，通过关键字 bind 与待测设计绑定。用断言描述属性特征时能清楚明确的定义任何属性^[9]。在本平台中大量使用断言来检测 ARINC629 控制器的属性是否符合设计需求。

图 5 为 ARINC629 控制器解码状态机转换示意图。仅以 SVA 对 ARINC629 接收模块解码状态机的属性描述中的一条为例，来

检验解码状态之间的转换及时序要求是否符合要求^[10]。

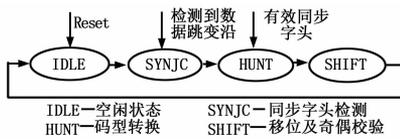


图 5 解码状态转换

在实际验证过程中, 当设计的属性与断言所描述的属性一致时, 断言不会有动作; 当设计的属性与断言所描述的属性不一致时, 即设计的运行偏离了最初的设计需求, 此时验证过程就会停止并生成相应的错误报告。针对 ARINC629 控制器的解码状态机转换, 编写相应的属性, 仅以 FSM_chk3 为例, 对解码状态机进行属性描述及覆盖率收集。其部分代码如下:

FSM_chk3: 状态 HUNT 和 SHIFT 不能直接由状态 IDLE 跳转得到。

```
// 属性描述
property p_forbid_trans3a;
@ (posedge Clk)
(((state == HUNT) || (state == SHIFT)) && ! Reset) | -
> past((state == IDLE) == 0);
endproperty
// 断言验证
a_forbid_trans3a;
assert property( p_forbid_trans3a );
// 覆盖率收集
c_forbid_trans3a;
cover property( p_forbid_trans3a );
```

3 验证结果分析

本平台在 QuestaSim10.0c 验证工具上对 ARINC629 控制器进行验证。

图 6 为语句和分支的覆盖率报告, 其中 Stmt% 为语句覆盖率, Branch% 为分支覆盖率。U_ARINC629_TC.v 为控制器的顶层模块, rxcontrol.v 为控制器的接收模块, txcontrol.v 为控制器的发送模块, shiftreg.v 为控制器的协议模块。图中顶层文件语句覆盖率达到 98.4%, 分支覆盖率达到 97%, 均未达到 100%, 查看其代码发现造成覆盖率达不到 100% 的原因是 ARINC629 控制器设计文件中存在 else 或 default 等语句。

(Recursive Coverage Aggregation) - Default						
File	Stmt	Stmts hit	Stmt %	Branch	Branches hit	Branch %
top						
xi						
tb						
U_ARINC629_TC	126	124	98.4%	66	64	97%
txcontrol	12	11	91.7%	14	13	92.9%
shiftreg	12	12	100%	8	8	100%
rxcontrol	15	15	100%	6	6	100%

图 6 语句和分支覆盖率

（上接第 1823 页）

参考文献:

[1] 谈乐斌. 火炮概论 [M]. 北京理工大学出版社, 2005.
 [2] 薄煜明, 等. 现代火控理论与应用基础 [M]. 北京: 科学出版社, 2012.
 [3] 程远斌. 某火炮系统设计与控制模型的建立 [J]. 四川兵工学报, 2002, 2: 17-21.

Cover Directives 窗口下观察到的覆盖率信息如图 7 所示, Count 为断言激活次数, Cmplt% 为断言指令的覆盖率。由图可知, 经验证后待测设计的各项功能均已达到其设计需求, 且实现率为 100%。

Name	Language	Enabled	Count	Cmplt %
/top/U_ARINC629_TC/asn/c_start1	SVA	✓	171	100%
/top/U_ARINC629_TC/asn/c_data1	SVA	✓	166	100%
/top/U_ARINC629_TC/asn/c_stop1	SVA	✓	107	100%
/top/U_ARINC629_TC/asn/c_start2	SVA	✓	149	100%
/top/U_ARINC629_TC/asn/c_data2	SVA	✓	144	100%
/top/U_ARINC629_TC/asn/c_stop2	SVA	✓	97	100%
/top/U_ARINC629_TC/asn/c_forbit_trans3a	SVA	✓	116	100%
/top/U_ARINC629_TC/asn/c_forbit_trans3b	SVA	✓	147	100%
/top/U_ARINC629_TC/asn/c_forbit_trans3c	SVA	✓	91	100%

图 7 功能覆盖率

4 结束语

通过分析验证结果可知, 该平台能够全面有效地对 ARINC629 控制器进行验证, 并确保后者完成了预期的功能。同时, 由于参照 ARINC629 协议, 并采用了可配置的测试环境及多种验证手段, 不但提高了该平台的可重用性, 且使得验证更加直观有效。平台架构及搭建方法具有可重用性, 更改验证平台中的数据生成模块即可用于其他类似设计的验证。

参考文献:

[1] Spear C. System Verilog for verification [M]. Synopsys Inc, 2006.
 [2] 朱 良, 杜承烈. 通用航空总线测试与分析软件的设计和实现 [J]. 微电子学与计算机, 2007, 24 (7): 125-127.
 [3] Aeronautical Radio, INC., ARINC Specification 629P1-5, Part1: Technical Description [Z]. Aeronautical Radio, INC., March 31 1999.
 [4] Gallon L, Juanole G, Blum I. Modeling and analysis of the ARINC Specification 629 CP MAC layer protocol [Z]. 1997.
 [5] 王继中, 樊晓樵, 黄小平. “龙腾”处理器 FPGA 验证平台的优化设计 [J]. 计算机测量与控制, 2009, 17 (1): 170-172.
 [6] Bergeron J, Cerny E, Hunter A, Andrew Nightingale. Verification Methodology Manual for System Verilog [M]. Synopsys Inc. & ARM. 2006.
 [7] SasanIman. Step-by-Step Functional Verification with System Verilog and OVM [M]. Springer, 2008.
 [8] Vijayaraghavan S, Ramanathan M. A Practical Guide for System-Verilog Assertions [M]. Springer, 2005.
 [9] Cerny E, Dudani S, Havlicek J, et al. The Power of Assertions in System Verilog [M]. Springer, 2010.
 [10] 周德新, 李 珊. OVL 断言验证库的 ARINC429 通讯板块设计验证 [J]. 计算机工程与设计, 2010, 31 (10): 2172-2174.

[4] 张红莲. 交直流调速控制系统 [M]. 北京: 中国电力出版社, 2011.
 [5] 郭 海, 吴益飞, 郭 毓. 车载发射转塔高性能随动系统智能控制策略研究 [J]. 计算机测量与控制, 2012, 20 (5): 1222-1228.
 [6] 金剑飞. 随动(改)系统设计 [D]. 南京: 南京理工大学, 2004.