

# 基于 Avalon-MM 的 I2C 总线控制器设计 设计及 VMM 验证

张晓龙

(上海大学 微电子研究与开发中心, 上海 200072)

**摘要:** 分析了 Avalon-MM 总线的架构以及 I2C 总线的传输方式, 提出了一种可行的 I2C 总线控制设计方案, 并说明了详细的实现过程; 设计的基于 Avalon-MM 总线的 I2C 总线控制器可复用 IP 核, 可以不加修改地应用于基于 Altera 公司产品的 SOPC 系统中; 使用 VMM 验证方法学构建了多层次可复用的验证平台, 对设计进行了全面的验证; 整个验证平台使用 systemverilog 语言实现, 仿真工具使用 VCS-MX200606; 最后仿真结果显示, 设计被完全地验证并达到 100% 覆盖率。

**关键词:** 总线接口; 控制器; 验证方法学; 覆盖率

## Avalon-MM Based I2C Bus Controller Design and Verification Using VMM

Zhang Xiaolong

(Microelectronic Research and Development Center, Shanghai 200072, China)

**Abstract:** In this paper, the architecture of Avalon-MM and the I2C bus protocol is analyzed. Then, a design of reusable I2C bus controller based on Avalon-MM bus and the detailed implement process is presented. This reusable IP core can be used in the SOPC system built on Altera company product without modification. A multi-level and reusable verification platform is implemented using VMM to verify this design. The whole verification environment uses the System Verilog language, and the simulation tool is Synopsys VCS-MX200606. The verification results indicate that design has been exactly checked out with 100% coverage.

**Keywords:** bus interface; controller; verification methodology manual; coverage

## 0 引言

现在半导体制造技术的发展已经可以实现一块芯片容纳几亿个晶体管, 这使得一块芯片中可以集成更多的系统功能, 并且引发了芯片设计方法学的变革。制造工艺的发展和对功能更强产品的需求使得 SOC 的设计成为主流<sup>[1]</sup>。传统的设计方法已经不能适用于这种规模复杂芯片的设计, 对于 SOC 设计, 设计复用一重用之前编写和验证过的代码一是唯一以可接受的代价完成如此复杂设计的方法<sup>[2]</sup>。现在设计人员的主要挑战是使完成的设计是可复用的和如何复用之前的设计。

使用标准总线接口的设计可以有效提高可重用性, 使用标准的接口可使设计易于理解和易于找到可用的 IP 核加快设计的进度<sup>[3]</sup>。另外, 针对此标准接口的验证 IP 核也可在多个项目中重用, 加速验证过程。使用标准接口的设计的功能模块也更容易在系统层次的集成, 因为 IP 接口信号与总线信号是统一的<sup>[4]</sup>。

随着芯片门数和复杂度的急剧增加, 芯片验证的工作量达到了整个芯片研发的 70%, 功能验证越来越成为设计过程中的瓶颈<sup>[5]</sup>。这催生了专门用于验证的编程语言 (systemverilog) 和一系列基于该语言的验证方法学的产生和发展, 例如 VMM, AVM, OVM, UVM 等。基于方法学的验证可通过多种机制显著提高验证生成率, 例如断言、抽象、自动化和重用<sup>[6]</sup>。

本文设计了一个 I2C 总线控制器<sup>[7]</sup>, 该控制器使用 Avalon-MM 标准接口, 可不加修改地应用于使用 Altera 公式产品构建的 SOPC 系统中<sup>[8]</sup>。然后应用 VMM 验证方法学为该设计搭建了一个可移植、重用、扩展、完全自动检查、具有层次化结构的验证平台, 使用受约束的随机激励生成和功能覆盖率等技术, 对设计进行了全面的功能验证, 并且达到 100% 的覆盖率。

## 1 设计描述

设计可以分为两个主要部分: I2C 协议产生部分和 Avalon-MM 总线接口部分。I2C 总线协议部分负责完成在 I2C 总线上按照协议完成数据的传输。Avalon-MM 总线接口部分联系系统处理器和 I2C 控制器部分, 处理器通过该接口完成对 I2C 传输数据和地址等信息的控制。设计的详细模块如图 1 所示。

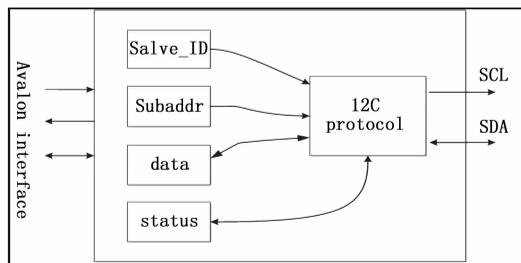


图 1 设计架构

### 1.1 I2C 协议产生器

I2C, 全称 Inter Integrated Circuit, 是由飞利浦公司开发的一种两线式串行总线。它用两根线, 时钟线和数据线, 实现数据的双向同步传输, 具有结构简单, 成本低等优点, 得到越

收稿日期: 2015-08-03; 修回日期: 2015-08-31。

**作者简介:** 张晓龙(1990-), 男, 河南开封通许人, 硕士研究生, 主要从事图像处理和集成电路的设计和验证工作。

来越广泛的应用。I2C 已经成为芯片间低速串行通信的事实标准，应用于各类消费电子、电子控制设备和各式传感器中。

I2C 总线采用简单的两线式结构，可以最小化引脚和 PCB 布线的开销。I2C 总线支持多主机结构，但在同一时刻只允许一个主机获得总线控制权，所以要通过仲裁来解决多个主机争抢总线的问题，以免发生数据冲突。每个从机都有唯一的 7 比特的地址码，主机依据此地址码选址从机。

I2C 定义了严格的数据传输时序，它以一个字节为单位进行数据传输，一次 8 比特数据传输结束后，接收方应发送一位响应位以确认数据接收成功。每次传输主机都要发送一位开始位和结束位来发起和结束一次数据传输。

1) 空闲状态：I2C 总线的空闲状态是时钟信号 SCL 的数据信号 SDA 同时处于高电平的状态，此时没有主机占用总线，由于上拉电阻的缘故，两根信号被上拉至高电平。

2) 起始位：I2C 总线定义在 SCL 为高电平时，SDA 由高电平跳到低电平为起始信号。通过发送起始位信号，主机通知从机开始数据传输。

3) 数据帧：在发送起始位后，主机开始从 SDA 逐位发送或接收数据帧，主机首先发送一个字节的从机地址码以寻址从机，该字节包含 7 比特的从器件地址和一位读写标志，1 代表读，0 代表写。需要注意的是，在数据帧传输期间 SDA 信号值只能在 SCL 为低时改变，只有在起始位结束时，SDA 才能在 SCL 为高电平时改变。

4) 相应位：主机或从机每发送完一个字节，需要对方法释放 SDA 的控制权来接收对方的响应。如果对方响应（SDA 为低电平）则可继续传输，如果不响应（SDA 为高电平）则应结束传输。

5) 结束位：I2C 总线规定当 SCL 为高时，SDA 由低电平变为高电平为结束信号。结束位的产生代表着一次传输的结束，总线回到空闲状态，直到下次起始位。

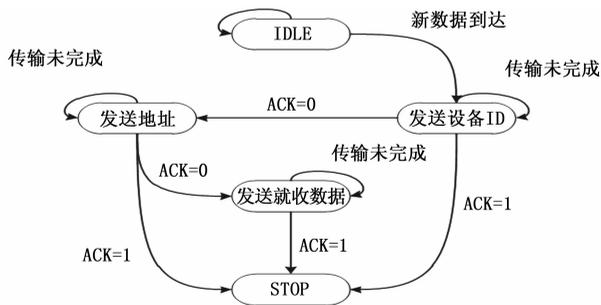


图 2 I2C 控制的状态机

### 1.2 Avalon-MM 总线接口

Avalon-MM 是 Avalon 总线的子集。Avalon 总线是 Altera 公司专门为可编程芯片上系统（System-On-a-Chip，以下简称 SOPC）而推出的一套片内总线系统，与 Nios 系列的处理器软核一起，二者构成了 Altera 公司 SOPC 解决方案中的核心部分。

Avalon-MM 全称是 Avalon Memory-Mapped，是一套交互式总线接口，主要用于多节点的互联，具有较好的数据交换特性和很高的总线带宽，由于是针对 SOPC 设计的，Avalon-MM 总线模块具有结构简单，采用全同步时序，以及可以灵活地配置等特点，其运行时钟、位宽、接口信号以及接口时

序等都可以灵活地选择或配置。

Avalon-MM 总线外设的控制或数据寄存器都可映射为处理器寻址空间内的一个地址。该设计中使用 4 个映射到地址空间的寄存器为 I2C 控制器提供所需的数据信息。寄存器 slave\_ID 用于指示要控制的从机编号和最后一位读写控制位。寄存器 sub\_addr 用于说明要访问的从机内部地址。寄存器 data 是要写入从设备或从从机接收到的数据。寄存器 status 指示控制器的当前状态。图 3 是本文设计的控制器选择的接口信号和对控制器内寄存器的读写时序。

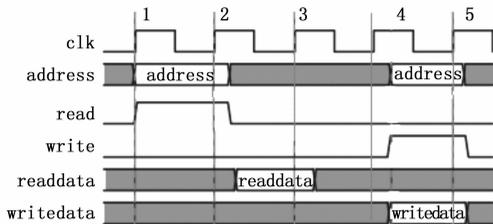


图 3 接口读取时序

### 1.3 接口与协议产生部分的连接

I2C 总线是低速总线，其时钟最大不超过 400 kHz。而外部的器件可能工作在很高的频率，比如 50 MHz 甚至更高。这样两个时钟域之间的信息交换可能会由于不能满足建立时间或保持时间而导致亚稳态的发生。为了避免亚稳态，使用了全同步式设计，所有的寄存器都由接口的高频时钟驱动，但为了实现 I2C 的低频传输，将使用分频得到的低频时钟作为参考时钟，通过脉冲边沿检测的方式，控制 I2C 状态机的转换，完成数据的低频传输。

Avalon 接口的 status 寄存器是状态寄存器，Avalon 可以写入该寄存器以控制 I2C 协议发生器，也可以读取该寄存器以了解 I2C 的当前状态。寄存器最后一位代表 I2C 协议发生器的闲忙状态，0 代表空闲，接口可以通过写入寄存器相应的值，并在 status 寄存器最后一位写入 1 来启动一次 I2C 传输，每次传输结束该位会被清零，可以接收下一次的传输。

## 2 验证平台

本文使用 VMM 验证方法学实现了一个分层验证平台，并设计了足够的测试用例来覆盖所有可能的边界条件。如图 4 所示，该验证平台包含激励生成器、驱动器、监视器、记分板和覆盖率统计等功能组件，它们协同工作完成对 DUT（design under test，待验证设计，这里指的是 I2C 控制器）的功能验证。

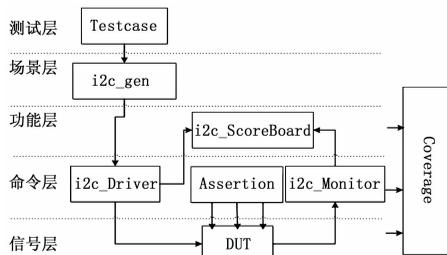


图 4 分层结构的验证平台

### 2.1 激励生成器

激励生成器 i2c\_gen 派生至 vmm\_xactor 类，用来随机

产生用于激励 DUT 的各种数据，并通过数据通道 (i2c\_trans\_channel) 传输给下层结构 i2c\_Driver。生成器产生的数据结构 i2c\_trans 派生至 vmm\_data，加入了工程相关的特定的数据结构和子程序。要进行一次 I2C 总线传输，需要知道从设备的标号、访问的从设备内部地址和传输的数据的等信息。但是不是每一个由生成器产生的这些信息都是有效的，所以使用受约束的激励生成器可以产生理想的数据，而不是之后分析并丢弃无效的数据。

随机生成器的目标是应该能生成完全验证特定设计所需的全部激励。不同的测试用例可能需产生不同的激励，所以生成器中使用了工厂模式，可以实现在 testcase 中改变要产生的目标数据，而不是修改 i2c\_gen 结构。

## 2.2 驱动器和监测器

在验证平台中，位于命令层的驱动器 (driver) 和监测器 (monitor) 是直接于 DUT 交互的功能组件，将上层的事务转化为信号层 DUT 引脚的实际激励。他们是基于事务 (transaction) 的验证环境的基础，更高的抽象级别是验证平台不在关注底层信号的细节，更容易复杂的功能。

接口 (interface) 是 systemverilog 中的一个重要数据结构，它是连接 DUT 中静态信号和验证环境中动态数据结构的纽带。I2C 控制器的 Avalon 总线接口被抽象成一个 interface 数据结构，该结构在驱动器实例化时被传进该验证组件内部，在驱动器内部以虚接口 (virtual interface) 的形式存在。之后，驱动器就可根据从上层组件接收到的事务并按照 Avalon 总线的时序对虚接口内的信号进行操作完成对 DUT 的驱动。

同样地，监测器也是通过虚接口跟 I2C 总线交互，通过监测这两根信号的变化，按照 I2C 协议来接收数据并做相应的响应。并将监测和响应的信息抽象成事务传输给上层结构。

## 2.3 自检结构和断言

自检结构记分板 (scoreboard) 是在运行时检查 DUT 响应是否正确的验证组件。记分板按顺序比较驱动器发给 DUT 的激励和监视器响应的数据，如果不相等则报出一个错误。

断言 (assertion) 是观察和定位设计中协议或规则错误的一种有效手段，当断言在仿真过程中失败，从报告的错误信息中很容易定位到错误所在。在本文中使使用断言检测 Avalon 总线协议和 I2C 协议的正确性。

## 2.4 功能覆盖率

功能覆盖率是衡量设计可信度和验证进度的重要指标，验证的目的是保证设计在实际的运行过程中的行为正确。设计规范对设计的行为进行详细的描述，验证计划从中抽象出相应的设计特性。验证平台为设计仿真出实际的运行环境，并从验证环境中收集数据以确定设计的哪些行为特性已经被仿真覆盖，以此估计验证的功能覆盖率。

功能覆盖率信息的收集使用驱动器回调函数 (callback method) 的方式实现，这能保证验证环境的灵活性。不同的测试用例可能要求验证组件表现不同的行为，回调函数使验证组件的功能是可控的和可配置的，而不必为每一个测试用例使用不同的验证组件。驱动器在每次向 DUT 发送激励后，执行回调函数并将激励信息作为参数传给回调函数，函数内的 covergroup 结构完成覆盖率信息的收集。

## 2.5 验证环境

验证环境类 i2c\_env 派生至 vmm\_env 类，作为验证平台

的顶层包装，它将各种验证组件封装在一起，并提供仿真过程控制的语法。验证平台的各种组件都会在该环境类中被创建和实例化，并在环境类的控制下完成仿真。为了避免竞争和冒险，仿真过程被分成许多个执行步骤，各个阶段完成指定的工作，这通过环境类中的各种虚函数实现。当 i2c\_env::run() 函数在 testcase 中被显式调用时，环境类中的以下虚函数会依次地被隐含地调用，完成各自的功能，最后完成仿真过程。

1) gen\_cfg(), 该函数产生随机的 DUT 和验证平台配置数据。用于配置 DUT 和验证平台的工作状态。

2) build(), 该函数创建各个验证组件，并用产生的配置信息配置验证环境，将各验证组件通过管道相互连接，以传输数据。

3) reset\_dut(), 该函数给 DUT 发送重置信号，是 DUT 进入初始状态，避免因为没有正确初始化导致的错误。

4) cfg\_dut(), 该函数用产生的配置信息配置 DUT，使其工作在合适的工作状态。

5) start(), 该函数使各个验证组件开始工作。

6) wait\_for\_end(), 该函数阻塞以等待测试的结束条件。

7) stop(), 该函数使各验证组件停止工作。

8) cleanup(), 该函数清理残留的数据，使仿真正常地结束。

9) report(), 该函数报告最终的仿真结果是失败还是成功，并关闭所有文件。至此一次仿真过程结束。

## 3 实验结果

使用 Synopsys 公司的 VCS 仿真工具对该控制器的设计和验证平台进行了仿真，VCS 首先对设计和验证环境的源代码进行编译，然后执行仿真过程。仿真的运行过程有验证平台的环境类控制。首先随机产生各种配置参数，创建验证组件的实例并用产生的配置参数配置验证平台。然后初始化并配置设计使之处于有效的状态，并开始启动各个验证组件，产生激励信息并驱动设计，同时观察设计的响应，根据自检结构的比对结果判断设计的功能正确与否。为了检测各种条件下设计的运行结果，读写方式和读写的地址空间都随机化。这样就可以产生读读、写写、读写、读写等不同的激励序列，对设计进行全方位的验证。等到功能覆盖率组件检测到预定义的覆盖率达到一定值，验证完成，就可以结束仿真过程。在结束仿真工作退出前，还有做些收尾工作，如停止各个验证组件的运行和报告仿真的结构等。这些工作对于仿真的正常运行和以后的统计和调试都是必不可少的。

仿真运行结果如图 5 所示。可以看到，覆盖率监视器报告的最后对 DUT 的验证达到 100% 的覆盖率，说明验证全面，对该控制器的所有可能的运行场景都进行了仿真。而自检结构记分板的报告显示共进行了 508 次的比对，并且全部比对正确。为了便于调试，在仿真过程中使用 verdi 工具提供的波形产生函数生成了波形，verdi 查看波形如图 6 所示。从图中也可以看到，Avalon 接口和 I2C 总线的时序都是满足协议标准的。综上所述，可以说明设计的 I2C 控制器功能正确，可以用于工程应用。

## 4 结论

基于标准总线的设计可以使一个复杂的 SOC 工程分为几

```

Normal[NOTE] on Coverage Callbck(cg) at          73723260:
Report coverage is                             100
Coverage is                                     100
Simulation PASSED on ./ (./) at                73724260 (0 warnings,
0 demoted errors & 0 demoted warnings)
Normal[NOTE] on I2C SCOREBOARD(sb) at         73724260:
Self-Checking result:                          508 Compare Correct; 0 Compare Err
or
$finish at simulation time                      73724260
VCS Simulation Report
Time: 73724260 ps
CPU Time:   21.340 seconds;      Data structure size:  0.4Mb
Sat Jul 25 17:57:06 2015

```

图 5 仿真运行结果

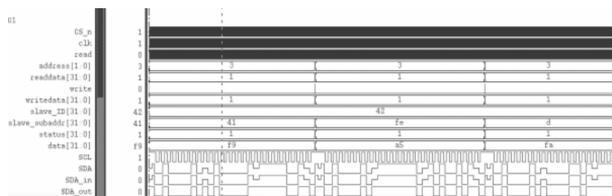


图 6 仿真波形

个部分，各部分可以独立地开发，最后集成在一起，并且可以复用已有的 IP 核以加速整个 SOC 的研发。本文设计的 I2C 总线控制器，可不加修改地应用于任何使用 Altera 公式产品构建的 SOPC 系统中。复杂设计给验证的挑战迫使验证在更高的

抽象级别上进行，本文基于 VMM 验证方法学搭建的验证平台在事务级运行，可显著提高验证的效率。验证平台的部分模块也可在系统级验证平台中复用。

参考文献:

[1] 钟 洪. 基于 FPGA 的 SOC 设计研究 [J]. 机电元件, 2013 (3): 13 - 18.

[2] Keating M J, Bricaud P. Reuse Methodology Manual for System-on-a-Chip Designs [M]. System-on-a-Chip Designs. Springer-Verlag GmbH, 2007: 486 - 490.

[3] Bergeron J. Verification Methodology Manual for SystemVerilog [M]. Springer, Berlin, 2006.

[4] 杨 鑫, 徐伟俊, 陈先勇, 等. Avalon 总线最新接口标准综述 [J]. 中国集成电路, 2007, 16 (11): 24 - 29.

[5] Wang J, Ji X H, Yang N X, et al. An Advanced Verification Environment Based on VMM [J]. Advanced Materials Research, 2013, 655 - 657: 900 - 904.

[6] 冯星宇, 黄 新, 颜学龙. 支持 AVALON 总线协议的 SPI 通信设计实现 [J]. 国外电子测量技术, 2013, 32 (3): 66 - 70.

[7] Semiconductor P. The I2C-Bus Specification Version2. 1 [EB/OL]. 2000; [http://www.semiconductors.philips.com/acrobat/various/I2C\\_BUS\\_SPECIFICATION\\_3.pdf](http://www.semiconductors.philips.com/acrobat/various/I2C_BUS_SPECIFICATION_3.pdf), 2000.

(上接第 146 页)

4 设计总结

本文开展射频识别技术 (Radio Frequency Identification Technology) 在汽车防盗装置中的研究，提出的射频识别装置有两部分组成，分别是汽车启动顶部的应答器和内部的阅读器构成，对整个装置的硬件和软件系统进行设计开发，首先搭建硬件系统的整体框架，重点对射频卡读写电路进行设计，包含射频电源和外围元件的选择以及射频电路的设计，软件系统设计方面，重点研究了读卡 and 写卡软件设计，给出了相应的时序图和程序流程图，达到了射频程序可靠读写的目的。本文给出的装置避免了市场上使用电池遥控装置，有效地实现汽车防盗的目的。

参考文献:

[1] 耿淑琴, 郭 跃, 王秀玲, 等. 一种低功耗液晶显示的便携式射频识别读写器系统 [J]. 电子器件, 2013 (01): 13 - 16.

[2] 丁开盛, 张学渊, 梁雄健. 通信网可靠性的定义及其综合测度指标 [J]. 通信学报, 2011 (10): 26 - 31.

[3] 张 帆, 孙 璇, 马建峰, 等. 供应链环境下通用可组合安全的 RFID 通信协议 [J]. 计算机学报, 2011 (10): 23 - 27.

[4] 杜永乾, 庄奕琪, 李小明, 等. 低功耗 UHF RFID 射频/模拟前端解决方案 [J]. 华中科技大学学报 (自然科学版), 2014 (09): 66 - 69.

[5] 欧阳宏志, 王新林, 朱卫华, 等. 基于 RFID 技术的网络式汽车安防系统的设计 [J]. 计算机测量与控制, 2011 (07): 43 - 48.

[6] 高 瞻, 胡向东. 基于 CORBA 的 RFID 仓储信息系统集成研究 [J]. 重庆邮电大学学报 (自然科学版), 2010 (04): 67 - 69.

[7] 李 宁, 黄银龙, 王占斌, 等. 双卡技术在泊车系统中的实现研究 [J]. 通信技术, 2011 (01): 81 - 86.

[8] 倪 龙, 和军平, 林廖军. 交互式无线汽车智能钥匙系统设计 [J]. 计算机测量与控制, 2010 (05): 98 - 103.

[9] 李翔翎. 用 VB6.0 实现微机与单片机之间的串行通信 [J]. 科技导报, 2013 (02): 38 - 43.

[10] 魏小微, 曹志刚. 低信噪比下数字幅度调制的调制进制快速识别 [J]. 清华大学学报 (自然科学版), 2013 (01): 61 - 67.

更 正

2015 年第 12 期第 4002 页《基于 WiFi 和虚拟仪器的噪声监测系统的设计》一文中，表 1 更正为：

表 1 某工程项目噪声监测结果

测点 编号	主要声源	测量值(分贝)			
		第一天		第二天	
		昼间	夜间	昼间	夜间
1	环境噪声	48.4	44.6	49.2	43.9
2	环境噪声	49.4	42.7	50.5	41.5
3	环境噪声	51.3	44.3	52.1	44.6
4	环境噪声	52.0	45.0	51.6	44.4
5	环境噪声	50.3	41.5	49.8	41.0
6	环境噪声	49.7	40.6	48.8	39.2
7	环境噪声	48.9	39.5	49.5	40.0
8	环境噪声	50.6	41.1	50.0	40.9