

# 基于 VMT 的 DDR-SDRAM 控制器功能验证

高 夫, 杜学亮, 金 西

(中国科技大学物理系微电子学教研室, 合肥 230026)

**摘 要** 介绍了一种基于验证模型技术(VMT)的 DDR-SDRAM 控制器的功能验证方案。该方案完成了 DDR-SDRAM 控制器对 DDR-SDRAM 模型的读写以及 AHB 2.0 协议的兼容性验证。VMT 的使用加快了验证平台的搭建和验证用例的编写。通过分析自动校对结果、仿真波形和覆盖率报告, 实现控制器功能验证的快速收敛。FPGA 原型验证进一步证明了该方案的可行性。

**关键词**: 验证模型技术; DDR-SDRAM 控制器; VIP 技术

## Functional Verification Based on VMT for DDR-SDRAM Controller

GAO Fu, DU Xue-liang, JIN Xi

(Institute of Microelectronics, Department of Physics, University of Science and Technology of China, Hefei 230026)

**【Abstract】** This paper introduces a solution for functional verification of DDR-SDRAM controller. The solution, based on VMT, completes the writing & reading operations happened between DDR-SDRAM controller and DDR-SDRAM model. The compatibility with AHB 2.0 specification is also verified. Using VMT accelerates building the testbench and writing the testcases. Fast convergence of functional verification is implemented by analyzing the auto-check results, simulation waveform and coverage report. The verification based on FPGA proves the feasibility of the solution.

**【Key words】** Verification Modeling Technology(VMT); DDR-SDRAM controller; VIP

AMBA(Advanced Microcontroller Bus Architecture)是当前SoC设计中的主流总线架构, AMBA部件验证是SoC验证的基础之一<sup>[1]</sup>。Synopsys公司基于验证模型技术(Verification Modeling Technology, VMT)开发了AMBA部件验证的一系列VIP(Verification IP), 可以很好地满足的验证需求, 实现重用, 缩短了SoC设计中AMBA总线部件的功能验证周期<sup>[2-3]</sup>。

### 1 VMT 的特点和应用

#### 1.1 VMT 的特点

Synopsys公司开发的兼容AMBA 2.0总线协议的VMT包括4个模块: Master, Slave, Monitor, Bus, 各个VMT模块可以通过函数set\_config\_param()来定义各自的参数, 比如数据总线的宽度、Monitor模块需要监控哪些Master和Slave等, 以配合不同AMBA总线部件的验证需要。这些模块在验证过程中的所有动作都以文本的方式保存在相应的目录之下, 这对定位错误非常有用。

图1描述了VMT模块和Testbench以及HDL或VERA仿真器的相互关系<sup>[4]</sup>。

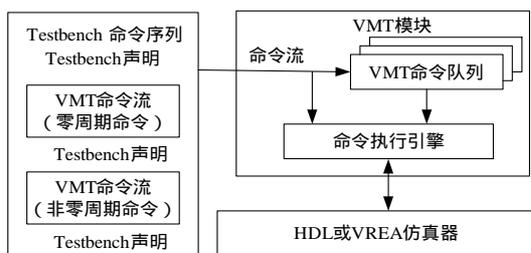


图1 VMT命令执行流程

在图1中, VMT模块接收来自Testbench中不同命令流的命令, 这些命令执行的周期数是不一样的, 可以是0周期(如设置VMT属性的命令), 也可以是1个或几个周期(如读

写命令)。VMT模块根据命令的不同属性将命令写入命令队列然后从队列中依次取出命令送入命令执行引擎模块, 或直接将命令送入命令执行引擎模块; 命令经命令执行引擎模块解析后, 由HDL或VERA仿真器实现仿真验证。

#### 1.2 VMT应用要点

随机验证是VMT一项非常重要的应用功能, 为验证的多样化提供了一个功能完善的平台, 加速了验证覆盖率的收敛。编写随机验证程序的流程如下: (1)定义需要应用随机验证的命令。(2)指定随机验证中需加的约束, 如地址范围、数据类型和比特宽度、猝发类型等。(3)确定命令队列和同一队列中各命令的相互关系。(4)将设计好的命令队列与命令队列或命令队列与单独的命令按先后关系在验证平台中组合串联起来。(5)把编写好的随机负载用于单独的命令或命令队列。

以下就是一个非常典型的约束例子:

- (1)XFER\_TYPE READ=40%, WRITE=40%, IDLE=\*;
- (2)BURST\_TYPE SINGLE=33%, INCR=\*;
- (3)NUM\_BEATS 1:4=5, 5:8=1;
- (4)ADDRESS 32'h0:32'h1fff=66%, 32'h2000:32'h3fff=\*。

在上面的例子中, 约束(1)表示在传输过程中, 读操作占40%, 写操作占40%, 其余为空操作; 约束(2)定义了33%的猝发(burst)传输是单个(single)类型的, 67%为递增(incremental)类型; 约束(3)表示猝发传输操作总数的5/6是1~4beats, 1/6是5~8beats; 约束(4)定义了地址的变化范围, 即在传输过程中产生的地址将有66%是从32'h0至32'h1fff, 34%是从32'h2000至32'h3fff。

VMT的应用范围主要在功能验证这一阶段, 尤其适合于

**作者简介**: 高夫(1982-), 男, 硕士研究生, 主研方向: SoC设计与验证, VLSI芯片设计; 杜学亮, 博士研究生; 金西, 副教授  
**收稿日期**: 2007-02-28 **E-mail**: jinxi@ustc.edu.cn

验证 SoC 中具有 AMBA 总线接口的 IP 核或自行开发的一些具有 AMBA 总线接口的比较通用的模块。在目前的 SoC 设计中,很大一部分使用的都是硬核形式的处理器(如 ARM 系列),因此功能验证阶段就无法借助于处理器,此时 VMT 中的 Master 模块可以代替处理器控制其他的 AMBA 总线部件,而不至于使功能验证被拖延,而且 VMT 具备非常完善的总线接口协议,可以充分验证总线部件的总线接口是否满足协议的要求。

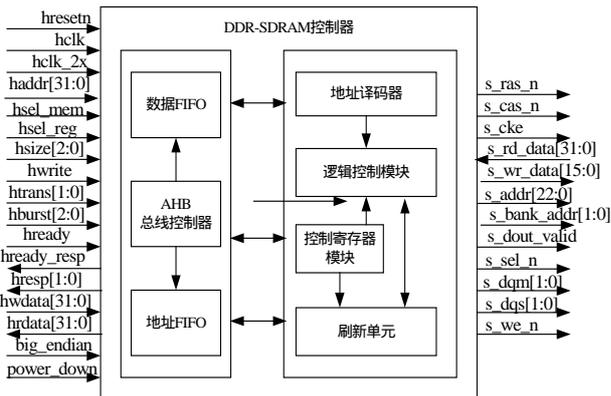
## 2 DDR-SDRAM 控制器的验证

### 2.1 DDR-SDRAM 控制器结构

当前 SoC 设计在外部 Memory 的选取上,很大一部分都倾向于 DDR-SDRAM,尤其在高速 SoC 设计,比如音视频处理 SoC。DDR-SDRAM 控制器是 SoC 设计中使用频率极高的一个模块。DDR-SDRAM 控制器兼容 AMBA AHB 2.0 总线接口协议, AHB 数据和地址总线均为 32 位宽,支持各种猝发读写,可配置大头(big-endian)和小头(little-endian)两种操作模式,通过配置各时序寄存器可支持不同厂家的 DDR-SDRAM 芯片,提供自动刷新(auto-refresh)功能,可配置 DDR-SDRAM 存储器起始地址和容量。

DDR-SDRAM 控制器的具体结构如图 2 所示,具有 AMBA AHB 接口和 DDR-SDRAM 存储器接口。

图 2 DDR-SDRAM 控制器结构



AHB 总线控制模块解析 AHB 控制信号,并操控数据 hwdata 和地址 haddr 分别进入数据 FIFO 模块和地址 FIFO 模块;地址 FIFO 模块除了缓冲地址,还包含猝发操作信息、读写标志位以及读写操作针对的是寄存器还是其标志位;地址译码模块将地址 FIFO 依次送出的地址译码以输出 s\_addr, s\_bank\_addr 和 s\_sel\_n 等信号;控制寄存器模块负责控制寄存器的读写及相关的控制操作,将寄存器中包含的信息传递给逻辑控制模块和刷新单元;逻辑控制模块根据 AHB 总线控制模块解析的结果将 AHB 控制信号转换为 DDR-SDRAM 存储器的控制信号,如 s\_ras\_n, s\_cas\_n, s\_we\_n, s\_cke, s\_dqm, s\_dqs 等;刷新单元负责根据控制寄存器的内容计算刷新频率,并通知逻辑控制模块,使其按照一定的时间间隔输出刷新控制信号。

### 2.2 VMT 参数设置

根据图 2 所示 DDR-SDRAM 控制器的接口,通过在 Testbench 中调用 VIP 中的 set\_config\_param()等函数,设置 VMT 的相关参数,参数的设置情况如表 1 所示。由于控制器有两个片选信号 hsel\_mem 和 hsel\_reg,因此将其看作 2 个 slave。

表 1 VMT 参数设置

参数名	值	实现函数
DW_VIP_AMBA_MASTER_PRESENT_PARAM	3	set_config_param
DW_VIP_AMBA_SLAVE_PRESENT_PARAM	2	set_config_param
DW_VIP_AMBA_HDATA_WIDTH_PARAM	32	set_config_param
DW_VIP_AMBA_HADDR_WIDTH_PARAM	32	set_config_param
DDR-SDRAM 分配的存储空间	64 MB	configure_memmap
存储器的地址空间单位	byte	set_config_param
是否支持猝发读写	是	set_config_param
读写转换之间是否插入 idle 态	否	set_config_param

## 3 验证平台及验证用例

### 3.1 验证平台

参考图 2,搭建基于 VMT 的 DDR-SDRAM 控制器验证平台如图 3 所示。由于 VMT 中有读取数据然后自动校验的函数 read\_expect(),因此省去了自动校验的模块,缩小了验证平台的规模。该验证环境中,除了 Dummy master,还有 Master1 和 Master2,这是为了模拟两个 Master 交替对 DDR-SDRAM 控制器进行操作的情况。DDR-SDRAM 模型是可以模拟 DDR-SDRAM 存储芯片的验证模型,可从生产厂家免费获得。

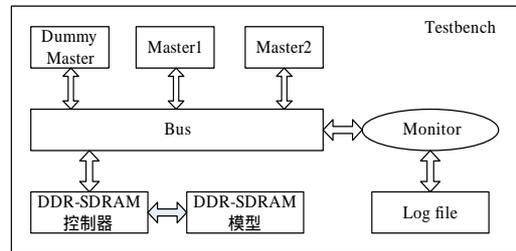


图 3 DDR-SDRAM 控制器验证平台

由于 DDR-SDRAM 存储器实际运行时时钟频率非常快,因此在 DDR-SDRAM 控制器与 DDR-SDRAM 模型连接时要做必要的信号处理。s\_dq 是 DDR-SDRAM 模型的 16 位双向数据线,在其与 s\_rd\_data 和 s\_wr\_data 连接时需要做双向释放处理,同时需要处理的还有双向信号 s\_dqs。具体的连接方法如图 4 所示。

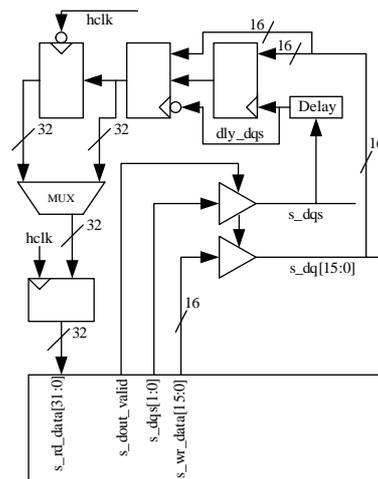


图 4 DDR-SDRAM 控制器接口处理

### 3.2 验证用例

为了全面地验证 DDR-SDRAM 控制器对 AMBA AHB 协议的兼容性和对 DDR-SDRAM 存储器的读写功能,本文所用验证用例将充分地考虑以上两方面。VMT 提供了足够完备的操作函数,编写验证用例过程中只要调用相应的 VMT 函数即可,这无疑加快了验证进程。验证用例包括:

(1) AHB 单字 1B, 23B, 4B 的读写,遍历 DDR-SDRAM

model 所有地址，读取时用 read\_expect()函数自动校对。

(2)AHB 不定长猝发读写，遍历 DDR-SDRAM model 所有地址，读取时用 read\_expect()函数自动校对。

(3)AHB 定长猝发(包括 burst=4, 8, 16)读写，遍历 DDR-SDRAM model 所有地址，读取时用 read\_burst\_expect()函数自动校对。

(4)改变 DDR-SDRAM 控制器中的控制寄存器，使 cas\_latency 分别等于 2, 2.5, 3。每种情况都要重复步骤(1)~步骤(3)。

(5)为了提高覆盖率，进行随机验证。随机约束定义 master 发出的传输类型和地址变化范围，并且通过函数 new\_random\_payload()为数据总线产生随机数。

(6)master1 和 master2 交替对 DDR-SDRAM 控制器进行读写操作，模拟实际 SoC 设计中不同 AMBA 部件对 DDR-SDRAM 控制器的控制请求。

限于篇幅，本文给出关键的验证程序，当 cas\_latency=3 时验证用例 1 和用例 3 的程序：

```

//验证用例 1
for(i=0; i=i+4; i<2^25)
begin
master1.write(`VMT_DEFAULT_STREAM_ID, i, 32'h5555_5555,
simpleIncrAttr, m1bufHandle[0]); //写操作
master1.read_expect(`VMT_DEFAULT_STREAM_ID, i, 32'h
5555_5555, simpleIncrAttr);
//读操作并自动校对
end
master1.block_stream(`VMT_DEFAULT_STREAM_ID, 0, status);
//建立命令流
//验证用例 3,
for(j=0; j=j+16; j<2^25) // burst=4 遍历 DDR-SDRAM model 写
begin
master1.new_burst_buffer(4*4, `VMT_MEM_PATTERN_INCR,
j*4/16, 4, m1bufHandle[j/32]); //建立新的猝发缓冲区
master1.set_burst_buffer_xfer_attr(m1bufHandle[j/16], 1, xfer
Attr[0]);
//设置猝发缓冲区属性
master1.write_burst(`VMT_DEFAULT_STREAM_ID, j, 4, m1buf
Handle[j/32], m1returnHandle[0]); //猝发写
end
for(j=0; j=j+16; j<2^25) // burst=4 遍历 DDR-SDRAM model 读
master1.read_burst_expect(`VMT_DEFAULT_STREAM_ID, j, 4,
m1bufHandle[j/16]);
//猝发读并自动校对
master1.block_stream(`VMT_DEFAULT_STREAM_ID, 0, status);
//建立命令流

```

### 3.3 验证结果评估

图 5 是 burst=4 的猝发写过程的示意图。在图 5 中，t1 时刻，AHB 总线往地址 31'h0000\_1000 写第 1 个数据 32'h1234\_5678，根据 AHB 总线协议，所写数据 32'h1234\_5678 滞后一个周期；t2 时刻，DDR-SDRAM 行地址为 1 的页存储空间被激活；t3 时刻，该页列地址自 16'h1000

开始的存储空间被激活；在 t4, t5 时刻，32'h1234\_5678 被分割成 2 个 16 bit 的数 16'h5678 和 16'h1234，这 2 个数据按先后顺序分别被写入 DDR-SDRAM 行地址为 1，列地址为 16'h1000 和 16'h1001 的存储空间中。

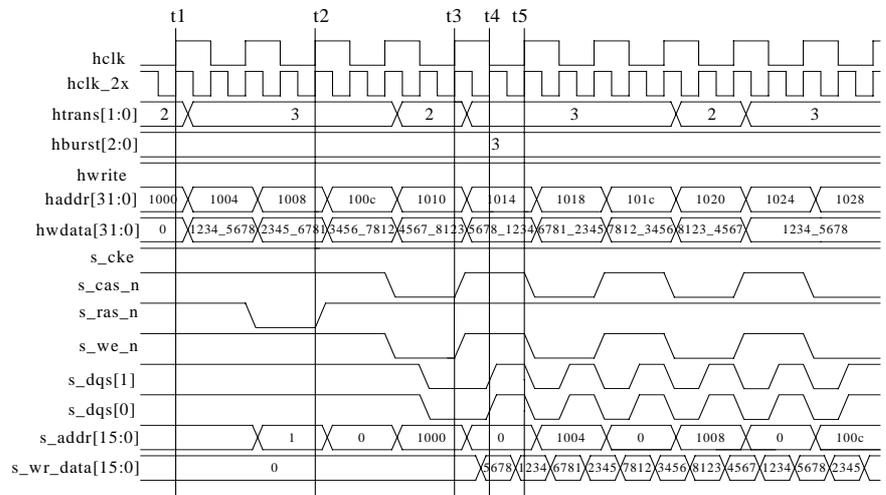


图 5 burst=4 时猝发写过程波形

经过仿真验证，得到覆盖率报告如表 2 所示。表 2 的覆盖率报告表明验证用例是满足要求的，充分、全面地验证了 DDR-SDRAM 控制器的功能。

表 2 覆盖率报告 (%)

行覆盖率	条件覆盖率	分支覆盖率	状态机覆盖率	信号翻转覆盖率
100	95.7	96.5	100	81.5

在整个验证过程中，验证平台的搭建从熟悉 VMT 文档到完成，1 人仅仅用了不到 1 周半的时间，编写验证用例以及进行错误检测用了 1 个月的时间。根据笔者所做实验，如果图 3 验证平台中的 Master，Bus 和 Monitor 模块自行设计，光是搭建验证平台就需要 3 周的时间，在该平台上验证 DDR-SDRAM 控制器则花了 1 个半月的时间，验证时间的增加是因为验证过程中还要纠正自行设计的 Master，Bus 和 Monitor 模块对 AHB 总线协议的一些错误理解以及由此引起的错误定位误差。由以上分析可以看出，基于 VMT 的验证周期是很有优势的。

### 3.4 FPGA 原型验证

FPGA 采用的是 Altera 公司的 EP2S60F1020C5。处理器选用 Mips4Kc 处理器，由于 Mips4Kc 处理器没有 AHB 接口，因此需要 Bridge 模块进行接口转换。DDR-SDRAM 芯片型号为 K4H511638D，利用 QuartusII 工具综合、布局布线后，得到资源利用报告，如表 3 所示。

表 3 DDR-SDRAM 控制器资源使用

资源	使用情况
ALUTs	2 398
registers	1 181
ALMs	1 273
LABs	208

表 3 中列出了 DDR-SDRAM 控制器的资源利用情况。FPGA 原型验证的时钟频率为 100 MHz。根据上述验证用例 1~用例 4 编写验证软件代码，编译后将其下载至 Mips4Kc 的 RAM 中，让 Mips4Kc 依照软件验证 DDR-SDRAM 控制器。经 FPGA 验证，DDR-SDRAM 控制器功能是正确的。

(下转第 274 页)