

高等学校通用教材

Verilog SOPC

高级实验教程

夏宇闻 黄 然 ©等编著



北京航空航天大学出版社

高等学校通用教材

Verilog SOPC 高级实验教程

夏宇闻 黄 然 等编著

北京航空航天大学出版社

内 容 简 介

Verilog SOPC 高级实验教程是为学习 Verilog 语言之后,想在 FPGA 上设计并实现嵌入式数字系统的人们而专门编写的。本实验教程是《Verilog 数字系统设计教程》(第 2 版)的后续课程,是姊妹篇。本书通过由浅入深的 10 个实验,详细地介绍了 ModelSim 6.0 和 Quartus II 8.1 的操作步骤,扼要地介绍了 Quartus II 8.1 的主要设计资源和 SOPCBuilder 等工具的应用方法,并阐述了如何配合自己设计的 Verilog 模块和 FPGA 中的内嵌处理器 Nios II 等现成 IP 资源,设计并实现高性能嵌入式硬件/软件系统。本实验教程也可以作为集成电路设计专业系统芯片(SoC)前端逻辑设计和验证课程的实验教材。为了使阐述的内容更加具体,本教程中的每个实验均选用 Altera FPGA(型号为 Cyclone II EP2C35F672C8)实现,并在革新科技公司专业级实验平台 GX-SOC/SOPC 运行通过。

本书可作为电子信息、自动控制、计算机工程类大学本科高年级学生和研究生的教学用书,亦可供其他工程技术人员自学与参考。

图书在版编目(CIP)数据

Verilog SOPC 高级实验教程/夏宇闻,黄然等编著. — 北京:北京航空航天大学出版社,2009.9

ISBN 978-7-81124-882-1

I. V… II. ①夏…②黄… III. 硬件描述语言, Verilog—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 143172 号

Verilog SOPC 高级实验教程

夏宇闻 黄 然 等编著

责任编辑 金友泉

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn>, E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×1 092 mm 1/16 印张:17 字数:435 千字

2009 年 9 月第 1 版 2009 年 9 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-882-1 定价:33.00 元(内配光盘)

序 言

我们知道,经过仿真证明是正确的 RTL Verilog 模块可以通过 EDA 工具综合成逻辑网表,并通过布局布线工具与物理电路对应起来。因此,正确无误的 RTL Verilog 模块可以很方便地转换成与某具体工艺对应的物理电路。这就是为什么说 RTL Verilog 模块比固定电路具有更大灵活性的缘故。RTL 模块不但可以映射到不同工艺和原理的基本电路,而且可以通过 Verilog HDL 语言支持的参数,将配置的参数值传入 RTL 模块,从而产生灵活多变的系列物理电路。

在本实验教程中,将从设计简单的模块开始,将其参数化,又通过参数设置将其转换成一个规模较大的电路。可以通过综合工具将其转变成逻辑网表,然后由布局布线工具将网表转换成某种 FPGA 芯片或某种工艺的物理电路。在本实验教程中,为了方便起见,只将其转换成 Altera Cyclone II EP2C35 FPGA 实验板上能运行的物理电路。首先经过 ModelSim 仿真已被证明行为和时序都正确的 Verilog 模块,配置引脚后还需要进行一次布局布线,然后将所生成的物理电路文件下载到 Cyclone II FPGA 实验板上,通过硬件运行,再一次验证所设计的电路是正确的,这样就有了一个随时可以使用的 RTL 模块。逐个积累每个模块的设计资源,再将这些模块配合系统中已经有的宏模块构成很大的系统,从而完成具有自己知识产权的复杂芯片设计。由此可见,所谓 SoC 芯片前端的逻辑设计过程,其实只是在理解芯片(设计项目)的功能和原理基础上,将其分割成可以操作的多个模块,逐块加以实现和验证,最后合在一起在大型 FPGA 上验证的过程。

在 Altera Quartus II 8.1 的工具包中包含许多已经验证的宏模块(megacore),这些宏模块为设计者构建复杂数字系统提供了极大的便利。由于开发环境中已经有许多可以利用的资源,所以,合理地利用免费的或者需要付一定费用的技术资源能显著加快设计的进度,提高设计的质量。这对数字系统设计师而言,无疑是一个很重要的环节,使得我们能对商业化的设计资源进行估价,根据市场的大小、上市进度,以及设计经费的预算等因素,迅速地做出采用商业化 IP 资源的权衡和决策,并加以实施。这些能力是 SoC 设计师应该具备的。

在本实验教程中,第 1 讲至第 5 讲和实验是为了介绍 ModelSim 仿真工具和 Quartus II 综合工具的使用,以及基本的设计方法、RTL 功能仿真、时序仿真和硬件运行仿真的概念;第 6 讲到第 10 讲内容和实验是为学习如何创建和利用现成的 IP 资源而专门设计的。

在本实验教程的后面,有很大一部分涉及 Nios II CPU 核的使用。任何复杂的数字系统都离不开负责处理人机界面、数学计算、系统内存管理、进程管理,和外设管理等基本操作的 CPU。学会利用 Altera Quartus II 的 SopcBuilder 工具,合理地配置系统所需要的嵌入式处理器核,并与自己设计的模块结合而构成一个完整的系统硬件架构,再配置合理的操作系统,并编写应用程序对于 SoC 系统的前端设计都是非常重要的。

SoC 芯片前端设计包括的内容非常广泛,数字电路的设计只是基础,而更多的内容涉及现成资源的合理配置和应用,特别是 CPU 资源的利用所涉及的面很广,不但有计算机体系结构、外围设备和操作系统的选择问题,还有运算速度和输入输出资源配套等问题。具体的

设计方法不但与想要实现系统的功能技术指标有关,还与各种 EDA 开发环境密切相关。因此编写这样的教材具有相当大的难度。作为教师只能为学生举几个具体例子,如配合想要设计的系统功能要求,做一定的合理划分,给每个模块分配确定的技术指标;找到并熟悉宏函数库中的有关 IP 核的配置和使用;定义必须自己设计的 IP 核;在设计中将这些 IP 核的实例构成完整的硬件系统;编写相应的软件,并在硬件平台上进行实际运行调试。本书中的许多实验是专门为以上教学目标而编写的。

负责 SoC 设计课题的指导教师,应该挑选有学习主动性的学生,利用寒暑假,集中一至三月以上的时间,或者利用本科生的毕业设计时间,就系统原理、体系结构、现成资源的应用、操作系统的选择和改进、应用软件程序编制等主题进行深入的讨论,分工协作共同完成设计任务,并对设计工作中的优缺点进行有针对性的讲评,鼓励有创造精神的学生。教师必须有意识地培养设计小组的团队精神,唯有这样才能加快培养我国新一代的 SoC 设计大师,争取我国 IC 设计界的业务水准在未来的 10 年内逐步接近和赶上世界先进水平。

我们还将陆续推出一些用硬件实现快速算法基础的原理性实验资料。其目的是让学生能通过实践透彻地理解把复杂算法转变成为特定硬件来提高运算速度的原理,从而理解算法性能的提高是与硬件设计密切相关的;灵活运用实验教程中的原理,便于将其应用到课题研究中去。特定算法的硬件研究开发是嵌入式系统应用的重要方面之一,也是关系到国防电子产品和高级消费类电子产品发展的关键技术。

2006 年我从北京航空航天大学退休后,先后受北京神州龙芯和巨数 IC 设计公司的邀请,到公司担任培训和顾问工作。本书的编写工作是在巨数 IC 设计公司支持下完成的。本实验教程的许多例子是从作者多年来为学校和公司培训 FPGA 设计工程师而专门编写的教材中精选的。2006 年初我与革新科技公司合作,带领四位硕士研究生杨雷、陈先勇、杨鑫、徐伟俊,根据革新科技公司提出的需求开发了一个由 FPGA 直接控制的 LCD 显示器和配套的演示软件。最后三讲中的 LCD 控制器硬件/软件接口实验是根据这个项目,为实现教学目标,经多次简化修改后定稿的。这个原创性的开发工作,对本实验教程的技术水准有至关重要的影响。作为本实验教程的合作者黄然、甘伟、陈岩和徐树,在学习期间帮助作者核对和整理了本教材中从第 1 讲到第 10 讲的全部实验资料。《Verilog SOPC 高级实验教程》是在这些实验材料的基础上,由我和所有这些年轻人经过多次共同讨论,互相启发,前后总共花费两年半时间编写成。初稿完成后,石家庄军械学院的满梦华老师认真审查了全书的实验,并把所有插图更新为 Quartus II 8.0 版本。特别应该感谢黄然工程师,他提出了许多宝贵的修改意见,把所有插图更新为 Quartus II 8.1 版本,再次核对了所有实验代码,做了相应的文字调整,补充了关于软件设计说明的小节,完成了所有思考题的标准答案,并且下载了必要的英文资料,刻录了供出版用的,带有所有课堂实验和思考题解答源代码和 Quartus II 8.1 工程文件的光盘中,方便了读者,提高了书的质量。全书的最后定稿由夏宇闻完成。

值此本书付梓之际,让我向巨数 IC 设计公司的商松董事长,邵寅亮技术总监,革新科技公司的沙时辉总裁,北京未名芯锐培训学校的曲韩宾校长和巨数 IC 设计公司的全体员工,以及所有为本书出版做出过贡献的人们表示衷心的感谢。

夏宇闻

北京巨数 IC 设计公司

参考文献

- [1] 夏宇闻. Verilog 数字系统设计教程(第2版)(M)北京:北京航空航天大学出版社,2008.
- [2] Samir Palnitka 《Verilog HDL, A Guide to Digital Design and Synthesis》2th Edition.
- [3] Altera 公司有关技术资料的网址:<http://www.altera.com/literature/lit-index.html>.
- [4] Altera 公司有关 Cyclone II FPGA 芯片系列的技术资料网址:<http://www.altera.com/products/devices/cyclone2/cy2-index.jsp>.
- [5] Altera 公司有关 Nios II 处理器 IP 核的技术资料网址:<http://www.altera.com/literature/lit-Nio2.jsp>.
- [6] Altera 公司有关 Quartus II 工具的技术资料网址:http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf.
- [7] 革新科技公司资料.GX-SOC/SOPC 开发实验平台手册.
- [8] Modelsim SE 6.0 User Manual. pdf (Modelsim SE 6.0 工具用户手册).
- [9] Modelsim SE 6.0 Tutorial. pdf (Modelsim SE 6.0 教程).
- [10] Avalon_specification. pdf (Avalon 接口规范).
- [11] Nios II_processor_reference. pdf (Nios II 处理器参考手册).
- [12] Nios II_software_developer. pdf (Nios II 软件开发者手册).
- [13] Volume 1 Design and Synthesis. pdf (Quartus II 设计和综合).
Volume 2 Design Implementation and Optimization. pdf (Quartus II 设计实现和优化).
Volume 3 Verification. pdf (Quartus II 验证).
Volume 4 SOPC builder. pdf (SOPC builder 工具使用手册).
Volume 5 Embedded Peripherals. pdf(SOPC builder 外设).

为了方便读者查阅,以下几份技术资料已经收集到本书附带的光盘中:

有关 Altera FPGA 器件、IP 核、工具、设计方法等更多的技术资料请参考:

<http://www.altera.com/literature/lit-index.html>

目 录

第 1 讲 ModelSim SE 6.0 的操作	1
1.1 创建设计文件的目录	2
1.2 编写 RTL 代码	3
1.3 编写测试代码	3
1.4 开始 RTL 仿真前的准备工作	4
1.5 编译前的准备、编译和加载	5
1.6 波形观察器的设置	5
1.7 仿真的运行控制	5
总 结	6
思考题	6
第 2 讲 Quartus 8.1 入门	8
2.1 Quartus II 的基本操作知识	8
2.2 Quartus II 的在线帮助	9
2.3 建立新的设计项目	9
2.4 用线路原理图为输入设计电路	12
2.4.1 图块编辑器的使用	12
2.4.2 线路原理图文件与 Verilog 文件之间的转换	16
2.5 编译器的使用	16
2.6 对已设计的电路进行仿真	19
2.7 对已布局布线的电路进行时序仿真	20
总 结	21
思考题	21
第 3 讲 用 Altera 器件实现电路	23
3.1 用 Cyclone II FPGA 实现电路	23
3.2 芯片的选择	24
3.3 项目的编译	26
3.4 在 FPGA 中实现设计的电路	27
总 结	35
思考题	35

第 4 讲 参数化模块库的使用	37
4.1 在 Quartus II 下建立引用参数化模块的目录和设计项目	37
4.2 在 Quartus II 下进入设计资源引用环境	37
4.3 参数化加法—减法器的配置和确认	38
4.4 参数化加法器的编译和时序分析	43
4.5 复杂算术运算的硬件逻辑实现	43
总 结	44
思考题	45
第 5 讲 锁相环模块和 SignalTap 的使用	48
5.1 在 Quartus II 下建立引用参数化模块的目录和设计项目	48
5.2 在 Quartus II 下进入设计资源引用环境	49
5.3 参数化锁相环的配置和确认	49
5.4 参数化锁相环配置后生成的 Verilog 代码	53
5.5 参数化 PLL 的实例引用	57
5.6 设计模块电路引脚的分配	59
5.7 用 ModelSim 对设计电路进行布局布线后仿真图	60
5.8 Signal Tap II 的使用	62
5.8.1 Signal Tap II 和其他逻辑电路调试工具的原理	63
5.8.2 调用 Signal Tap II 的方法	63
5.8.3 Signal Tap II 的配置	64
总 结	70
思考题	71
第 6 讲 Quartus II SOPCBuilder 的使用	73
6.1 Quartus II SOPCBuilder 的总体介绍	73
6.2 SOPCBuilder 人机界面的介绍	73
6.3 将 Nios II 处理器核添加到系统	75
6.4 部件之间连接的确定	76
6.5 系统内存部件的确定及其在系统中的添加	77
6.6 系统构成部件的重新命名和系统的标识符	78
6.7 基地址和中断请求优先级别的指定	78
6.8 Nios II 复位和异常地址的设置	79
6.9 Nios II 系统的生成	79
6.10 将配置好的 Nios II 核集成到 MyNiosSystem 项目	81
6.11 用 Nios II 软件集成开发环境 IDE 建立用户程序	83
6.12 软件代码解释	87
总 结	88

思考题和实验	88
第 7 讲 在 Nios II 系统中融入 IP	91
7.1 Avalon 总线概况	91
7.2 设计模块和信号输入电路简介	92
7.2.1 LED 阵列显示接口的设计(leds_matrix. v)	92
7.2.2 按钮信号的输入(button. v)	98
7.3 硬件设计步骤	100
7.3.1 建一个目录放置设计文件	100
7.3.2 创建设计的组件	101
7.3.3 Nios II 系统的构成	105
7.3.4 对 Verilog 文件的归纳和编写设计项目的顶层文件	108
7.3.5 用 .tcl 文件对 FPGA 引脚的定义	110
7.3.6 对项目的编译	113
7.3.7 把编译生成的电路配置代码下载到 FPGA	114
7.4 软件设计步骤	114
7.4.1 建立软件程序目录并调用 Nios II IDE	114
7.4.2 程序的运行	118
总 结	118
思 考 题	119
第 8 讲 LCD 显示控制器 IP 的设计	121
8.1 LCD 显示的相关概念介绍	121
8.1.1 位图的基础知识	121
8.1.2 位图的尺寸	122
8.1.3 位图颜色	122
8.1.4 地址的线性、矩形选择	122
8.1.5 alpha 混合	122
8.1.6 TFT-LCD 彩色显示控制时序图	123
8.1.7 显示器控制接口(IP)知识产权核介绍	124
8.2 显示控制器 IP 核总体结构及其与嵌入式 Nios II 处理器核的关系	125
8.3 端口信号的说明	127
8.4 显示控制器 IP 核的基本操作	128
8.5 显示控制器 IP 寄存器的说明	128
8.5.1 寄存器总体介绍	128
8.5.2 控制寄存器组	129
8.5.3 时序寄存器组	130
8.5.4 背景层相关寄存器组	131
8.6 模块划分及模块功能简介	132

8.7	LCD IP 模块的测试	137
8.8	在 SOPC 系统中应用 LCD 显示控制器 IP 核	137
8.9	构建 SOPC 系统	145
8.10	引脚分配	151
8.11	软件开发	153
8.12	软件代码解释	158
	总 结	158
	思考题	159
第 9 讲	BitBLT 控制器 IP	161
9.1	图形加速及 BitBLT 相关概念介绍	161
9.1.1	位图和 BitBLT	162
9.1.2	调色板	162
9.1.3	颜色扩展	163
9.1.4	颜色键控	163
9.1.5	光栅操作	163
9.2	BitBLT 控制器 IP 介绍	164
9.2.1	BitBLT 控制器 IP 结构和系统结构框图	164
9.2.2	BitBLT 控制器 IP 寄存器说明	166
9.2.3	BitBLT 控制器 IP 模块说明	167
9.3	BitBLT 控制器 IP 使用示例	169
9.3.1	构建 SOPC 系统	169
9.3.2	引脚分配	173
9.3.3	软件开发	176
9.3.4	软件源程序	182
9.3.5	软件代码解释	189
	总 结	190
	思考题	191
第 10 讲	复杂 SOPC 系统的设计	192
10.1	本讲使用的主要组件简介	193
10.1.1	LCD 控制器	193
10.1.2	BitBLT 控制器	193
10.2	硬件设计步骤	193
10.2.1	Quartus II 工程的建立	193
10.2.2	在工程中加入 LCD 控制器和 BitBLT 控制器	194
10.2.3	Nios II 系统的构成	194
10.2.4	编写设计项目顶层文件	194
10.2.5	FPGA 引脚定义	194

10.2.6 编译和下载项目	194
10.3 软件开发	194
10.3.1 软件程序介绍	194
10.3.2 软件结构	195
10.3.3 软件源程序	195
10.3.4 软件代码解释	203
总 结	203
思 考 题	205
本书的结束语	206
附录 GX-SOC/SOPC 专业级创新开发实验平台	207
附录 1 GX-SOC/SOPC-DEV-LAB Platform 开发实验平台概述	207
附录 2 GX-SOC/SOPC-DEV-LAB Platform 创新开发实验平台简介	209
附录 3 GX-SOC/SOPC-DEV-LAB Platform 创新开发实验平台的组成和结构	210
附录 4 GX-SOPC-EP2C35-M672 Cyclone II 核心板硬件资源介绍	253
参考文献	259

第 1 讲 ModelSim SE 6.0 的操作

前 言

在《Verilog 数字系统设计教程》中我们学习了 Verilog 的语法,学习了如何用 Verilog 语言来描述可综合成实际电路的组合逻辑、时序逻辑、状态机和复杂数字系统,最后学习了如何用描述行为的 Verilog 测试模块对被设计的复杂电路进行全面的测试和仿真。但这并没有深入讲解设计过程中所使用的工具,以及如何借助于仿真和综合工具,逐步实现设计电路的细节。

在本讲中将通过一个简单的设计示例,首先介绍如何使用 ModelSim SE 6.0 工具进行功能仿真,然后在以后几讲中分别介绍 Quartus II 8.1 工具,以及如何在 FPGA(Altera Cyclone II EP2C35F672C)上实现数字逻辑电路,产生时序仿真必须的带延迟的门级模型。在此基础上,学生可以自己动手在开发平台上,下载 Quartus II 工具生成的电路构造信息,生成具体的物理电路,从而进行一系列实验。实验的安排从简单到复杂,循序渐进,直至利用商业化的 CPU 核和高级的外围 IP 核,并配合自己设计的 IP 核,在 FPGA 上实现极其复杂的数字系统的硬/软件设计。

为了使讲解更具体和更具有针对性,将选用网络上可以下载的,并在 PC 机上运行的 ModelSim 和 Quartus II 两个工具,以及性能稳定可靠的革新科技公司开发的 FPGA 平台,作为硬/软件运行的实验环境。同学们在掌握工具的基础上,阅读本教程,就可以自己动手逐个完成教程指定的示范实验,并按照思考题改进设计,在开发平台上运行自己设计的逻辑电路和系统。

这样的独立自主的实验活动,无疑能提高同学们的学习兴趣,培养团队精神和解决实际问题的能力。作者建议有数字系统设计教学任务的电子、计算机和控制工程专业的院系应该提供相应的实验室和设备,利用寒假和暑假,24 h 对学生开放。在完成教学任务的基础上,集中一至三个月以上的时间,鼓励有兴趣的学生独立完成整套实验,由负责教师进行讲评。只有采用启发式的教学方法,注重实际动手能力的培养,未来中国集成电路的前端设计人才会源源不断地涌现。

设计数字电路必须首先了解电路的功能,如果功能很简单,可以立即开始编写寄存器传输级(RTL)代码。例如要设计一个 2 选 1 多路选择器,就可以用 ModelSim 所带的文本编辑器或者任何一种文本编辑器编写一个模块,存入指定的目录。如果功能比较复杂,就需要认真分析,将其分割成可以操作的功能块,逐块加以解决。

为了使同学们入门容易,先从如下简单的例子开始,学习工具的使用步骤,然后通过循序渐进的实验练习,构建起一个比较复杂的数字系统。

1.1 创建设计文件的目录

设计工作的第一步是在计算机的文件系统中创建一个目录,目录的名称要反映设计的内容,例如 C:\vlogexe\book\muxtop。这里设计的模块是一个位宽可变的 2 选 1 多路器。为了达到设计目标,首先设计一个位宽为 1 的 2 选 1 多路器。单击 ModelSim 图标,随即弹出 ModelSim SE 6.0 的主窗口,如图 1.1 所示。

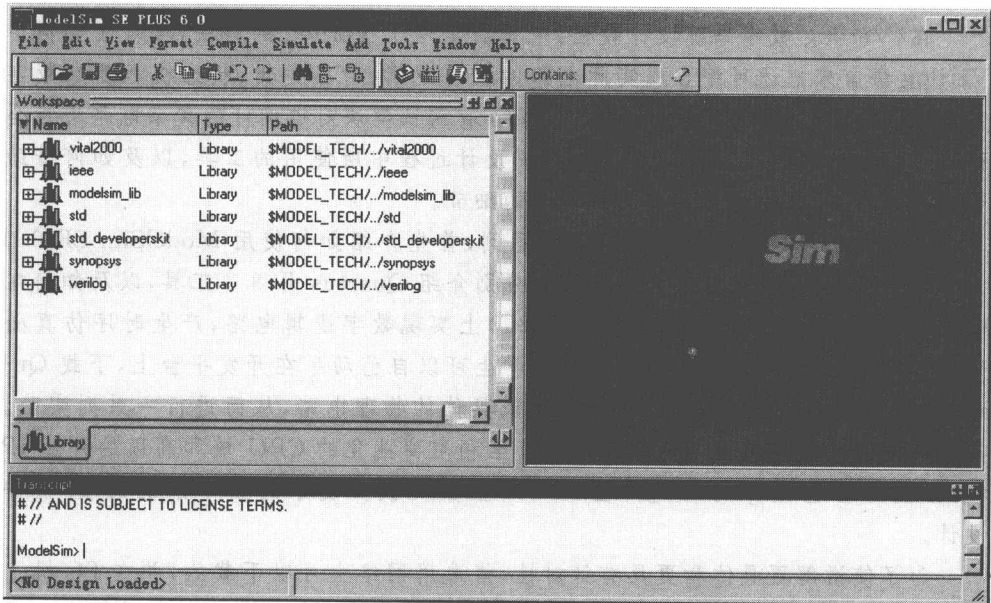


图 1.1 ModelSim SE 6.0 主窗口

主窗口分成三部分:工作空间(workspace)子窗口、主窗口内容和执行情况报告子窗口(Transcript)。单击子窗口右上角带箭头的符号,可以将子窗口脱离主窗口单独显示,再次单击又可以回到主窗口。子窗口的分布和大小可以通过鼠标调整,也可以通过单击主窗口主菜单栏的 Window 命令进行设置。

有些命令,必须按顺序访问两个或者两个以上的菜单。为了简单地说明命令的使用步骤,可约定使用:菜单 1>菜单 2>具体项 来表示选择操作命令的步骤。用户先在菜单 1 有关项上单击鼠标的左键,然后在此产生的下拉或弹出菜单 2 中单击具体项。举例说明如下:可以通过单击子窗口右上角的图标按钮将子窗口独立出来、放大、缩小或者关闭。而单击 Window>InitialLayout 命令就可以把分布很乱的子窗口恢复原始状态。大家可试验一下,如何正式开始整个设计过程。

单击主窗口中的命令序列:File>Change Directory...,在弹出的窗口上选择工作目录为:C:\vlogexe\book\muxtop,这样就把工作路径改到指定目录下了。然后单击命令 File>New>Source>Verilog,随即在主内容窗口上出现文本编辑窗口,设计者就可以开始编写 Verilog 代码。

1.2 编写 RTL 代码

[例 1] 设计一个位宽可变的 2 选 1 多路器。

学习过《Verilog 数字设计教程》的同学,可以很容易地编写出如下的 Verilog 代码:

```
module mymux(a,b,sel,out);
  `parameter width= 1;
  input sel;
  input [width- 1:0] a;
  input [width- 1:0] b;
  output [width- 1:0] out;
  reg [width- 1:0] out;
  always @ (* ) //Verilog 2001 支持用这种方式表达敏感列表
    if(sel)
      out= a;
    else
      out= b;
endmodule
```

将以上文件存入 C:\vlogexe\book\muxtop\mymux.v 文件中。再编写以下代码,m2 实例引用了上面已设计的模块 mymux,并将参数(parameter)width 在 m2 中重新定义为 2。在 m2 实例引用 mymux 时,用 #2 将新定义的参数 width=2 传入。完整的代码如下所示:

```
module muxtop(sel,a2,b2,out2);
  input sel;
  input [1:0] a2, b2;
  output [1:0] out2;

  mymux #2 m2(.a(a2),.b(b2),.sel(sel),.out(out2));

endmodule
```

将以上文件存入 C:\vlogexe\book\muxtop\mux_top.v 文件中。

1.3 编写测试代码

为了验证设计的 RTL 代码是否正确,还需要编写测试模块。测试 muxtop 模块的代码如下所示:

```
`timescale 1ns/1ns
`define period 100
module t;
  reg sel;
  reg clk;
  reg [1:0] a2, b2;
```

```

wire [1:0] out2;
initial
begin
    clk= 0;                //时钟值初始化
    sel= 1;
    # (1000* `period)MYMstop;    //等待 1 000 个周期后停止仿真
end
always# (`period/2)  clk = ~ clk;    //产生测试用时钟
always# (`period* 20)  sel = ~ sel; //产生选择信号的改变
always @ (posedge clk)
begin
    # 1  a2= {MYMrandom} % 4;        //位拼接符可以使 $ random 只产生正的随机数
    # 2  b2= {MYMrandom} % 4;        //产生 0~3 之间的随机数序列
end

muxtop m1(. a2(a2),. b2(b2),. out2(out2),. sel(sel));

endmodule

```

将以上文件存入 C:\vlogexe\book\muxtop\t.v 文件中。

1.4 开始 RTL 仿真前的准备工作

在进行功能仿真前,必须先建立一个 Library(库)来记录有关功能仿真的信息。命令序列为:File>New>Library,随即弹出一个对话框,键入 Library(库)的名字,随便起一个名字即可,但最好有含义,例如现在是做功能仿真,起名 RTLsim 比较合适。起完名后,在工作空间的子窗口就能见到生成一个空的 RTLsim 新目录。然后单击主窗口上的编译图标(类似几页纸上面一个蓝色的向下箭头),或者用命令序列:Compile>Compile...,随即弹出一个初次编译对话框,如图 1.2 所示。

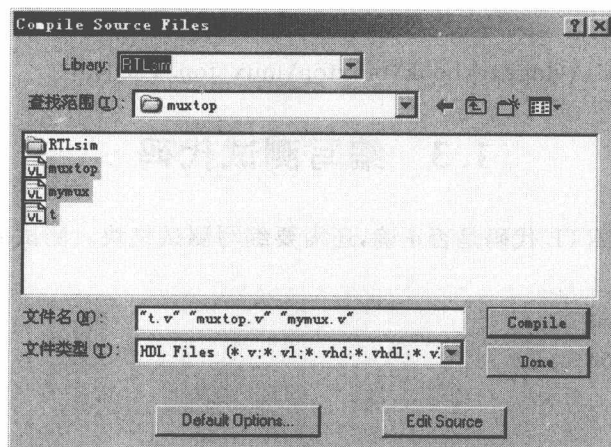


图 1.2 初次编译对话框

1.5 编译前的准备、编译和加载

编译前的准备、编译和加载步骤是:单击 Library 选择框的右侧,在出现的下拉菜单中选择 RTLsim,然后选择 mymux. v, muxtop. v, t. v,单击上述窗口中的 Compile,即可完成编译。随后在主窗口中的工作空间子窗口中的 RTLsim 目录中就会出现这三个文件的名字,说明编译顺利通过。在报告子窗口中也有信息表明编译顺利完成。如果编译出现错误,双击报告子窗口中出现的错误信息,便能自动地提示发生错误的程序行,帮助设计者发现错误。双击工作空间子窗口中 RTLsim 目录下的 t,随即就可以将编译后的代码加载到仿真器。必须注意 Library 的名称是你想要的(本例子中为 RTLsim),编译后的信息都记录在这个库中。如果加载成功,就可以准备波形观察器的信号设置。如果加载不成功,报告子窗口中将出现错误信息提示,必须认真分析错误信息,从而找到问题出在哪里。加载成功后,工作空间子窗口将自动从 Library 子窗口转移到 sim 子窗口。

1.6 波形观察器的设置

在工作空间(workspace)子窗口 sim 中,右击想要观察波形的模块 t,然后在出现的菜单上单击 Add>Add to Wave 项,就出现带有可观察信号的波形框图。若还需要观察其他模块的信号波形,可以用类似方法添加。

1.7 仿真的运行控制

单击波形观察器窗口上面(图案为文本右侧有一个蓝色双箭头)的 Run-All 图标,就可以将测试模块运行到结束,因为在测试程序 t 中,设置了控制仿真时间(注意:t 模块中,initial 块中的最后一句语句是 #1000 * period \$ stop;),所以,到指定时间仿真会自动停止,就可以显示仿真波形。也可以在主窗口下的 Transcript 子窗口下键入命令:run 100000。还可以用主窗口台头下的命令序列:Simulate>Run>Run-All,或者其他选择项。设计者可以根据实际情况,自行控制运行的时间延续时间。muxtop RTL 级 Verilog 模块的仿真波形如图 1.3 所示。

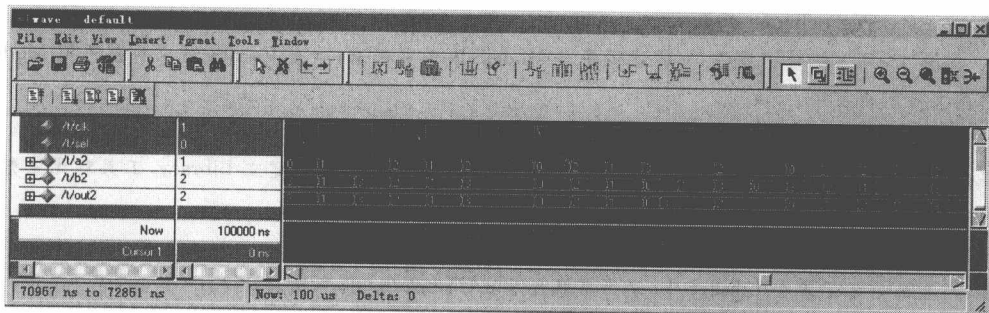


图 1.3 muxtop RTL 级 Verilog 模块的仿真波形

从波形图上可以清楚地看到当 sel 信号为 1 时,输出信号 $out2=a2$;而当 sel 信号为 0 时,输出信号 $out2=b2$ 。因此,这个 2 选 1 多路器的功能是正确的。由于测试信号是随机发生的,所以,可以测试很长的时间,设计者可以仔细地观察波形,发现问题就及时修改。波形观察器上显示的数据的格式可以单击信号名,然后通过波形观察器菜单栏命令 `format>Radix>unsigned` 来改变。默认情况下显示的是二进制数。波形内数据的格式可以有多种选择,也可以将波形显示的数字转变成波形幅度,总之变化是非常丰富的。必须认真阅读 Help 有关细节才能有更多的了解,多实践,就能丰富自己的实际操作经验,从而更加灵活地应用环境资源。

到目前为止,我们已经讲解了:

- (1) 怎样用 RTL Verilog 设计位宽可变的 2 选 1 多路器的基本方法;
- (2) 怎样编写行为模块对设计模块进行的测试,观察波形以进行功能验证。

在第 2 讲中,将讲解如何用 Quartus 进行综合,产生可以进行布线后仿真的 Verilog 模块,在这里仍旧用 ModelSim 对其进行布线后的 Verilog 模型进行仿真,以进一步确认设计的正确性。如果布线后仿真是正确的,就可以通过 Quartus II 工具将生成的逻辑配置信息下载到 FPGA 芯片中,利用革新科技公司的开发平台,试验实现电路的功能是否正确。

总 结

第 1 讲介绍的方法是最基本的,ModelSim SE 6.0 以上版本有非常强大的功能,同学们若想要全面地掌握这个工具,必须认真阅读与 ModelSim 工具有关的手册,更重要的是必须敢于实践,勤于实践,不断地摸索。读者可以利用这个例子,改动程序,故意让程序出错,然后观察系统的反应,用系统提示的错误信息,查找设计中的错误,随意做各种试验,以逐步了解每个窗口的各种命令和选择项的含义。

思 考 题

- (1) 为什么必须首先定义一个自己的目录来放置设计文件?这样做有什么好处?

答:为了与 ModelSim 原有的库相区别,创建一个自己的工作目录。

- (2) 如何将 ModelSim 环境中的路径转变到自己设置的目录?

答:方法一,File>change directory。

方法二,可以把带扩展名 .v 的源程序与 ModelSim 关联起来,单击 Verilog 源文件就可把 ModelSim 自动引入文件所在目录。

- (3) 为什么每次设计之前必须新建 Library?

答:ModelSim 的 Library 是用来记录编译过程中产生的信息,如果不建立 Library,工具就无法知道编译后的信息放在哪里。因而无法开始编译工作。

- (4) Verilog 模块编译时产生的信息记录在什么地方?

答:产生的信息记录在 Library 中,运行信息显示在 Transcript 窗口中。

- (5) 如何使用编译工具?有几种方式启动编译工具?

答:方法一,Compile 菜单;方法二,单击编译图标。

- (6) 为什么每次使用独立编译窗口时,必须核对一下 Library 是否正确?