

高等学校通用教材

Verilog

数字系统设计

Digital System Design **教程**

夏宇闻 编著

Tutorial



 北京航空航天大学出版社
<http://www.buaapress.com.cn>

Verilog 数字系统设计教程

夏宇闻 编著

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内 容 简 介

本书讲述了自 20 世纪 90 年代开始在美国和其他先进的工业国家逐步推广的利用硬件描述语言(Verilog HDL)建模、仿真和综合的设计复杂数字逻辑电路与系统的方法和技术。本书从算法和计算的基本概念出发,把复杂的算法逐步分解成简单的操作步骤,讲述最终如何由硬线逻辑电路来实现复杂的数字逻辑系统。

全书共分四部分,另加一个附录。第一部分为初级篇,共 8 讲;第二部分为中级篇,共 10 讲;第三部分为实践篇,共 12 个实验例选。最后一部分是 Verilog 硬件描述语言参考手册,即语法篇,供学习、查询之用。

书中突破传统的章节结构,以“学时”为单位,以“讲授”为形式,将 Verilog HDL 知识由浅入深、由简单到复杂、由典型到一般的循序渐进方法,一步步地传授给读者,使大学电子类及计算机工程类本科及研究生,以及相关领域的设计工程人员可以从掌握 Verilog HDL 技术,把我国的 IC 设计、复杂数字系统设计推向一个全新的高度。

本书可作为电子工程类、自动控制类、计算机类的大学本科高年级及研究生教学用书,亦可供其他工程人员自学与参考。

图书在版编目(CIP)数据

Verilog 数字系统设计教程/夏宇闻编著. —北京:
北京航空航天大学出版社,2003.7

ISBN 7 - 81077 - 302 - X

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

中国版本图书馆 CIP 数据核字(2003)第 019982 号

Verilog 数字系统设计教程

夏宇闻 编著

责任编辑 金友泉

责任校对 陈 坤

*

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(100083) 发行部电话:010 - 82317024 传真:010 - 82328026

<http://www.buaapress.com.cn>

E-mail: bhpess@263.net

河北省涿州市新华印刷厂印装 各地书店经销

*

开本:787 mm×1 092 mm 1/16 印张:30 字数:768 千字

2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷 印数:4 000 册

ISBN 7 - 81077 - 302 - X 定价:38.00 元

前 言

数字信号处理(DSP)系统的研究人员一直在努力寻找各种优化的算法来解决相关的信号处理问题。当他们产生了比较理想的算法思路后,就在计算机上用C语言或其他语言程序来验证该算法,并不断修改以期完善,然后与别的算法作性能比较。在现代通信和计算机系统中,对于DSP算法评价最重要的指标是看它能否满足工程上的需要。而许多工程上的需要都有实时响应的要求,也就是说应需要数字信号处理(DSP)系统在限定的时间内,如在几个ms甚至于几个 μs 内,对所输入的大量数据完成相当复杂的运算,并输出结果。这时如果我们仅仅使用通用的微处理器,即使是专用于信号处理的微处理器,往往也无法满足实时响应的要求。因此,不得不设计专用的高速硬线逻辑来完成这样的运算。设计这样的有苛刻实时要求的复杂的高速硬线运算逻辑是一件很有挑战性的工作,即使有了好的算法而没有好的设计工具和方法也很难完成。

近30年来,我国在复杂数字电路设计技术领域与国外的差距越来越大。作为一名在大学教授专用数字电路与系统设计课程的老师深深感到自己身上责任的重大。我个人觉得,我国在这一技术领域的落后与大学的课程设置和教学条件有关。因为我们没有及时把国外最先进的设计技术介绍给学生,也没有给他们创造实践的机会。

1995年我受学校和系领导的委托,筹建世行贷款的电路设计自动化(EDA)实验室。通过近10年的摸索、实践,我们逐步掌握了利用Verilog HDL设计复杂数字电路的仿真和综合技术。在此期间我们为航天部等有关单位设计了13万门卫星信道加密用的复杂数字电路,提供给他们经前后仿真验证的Verilog HDL源代码,得到很高的评价。在其后的几年中又为该单位设计了10万门卫星下行信道RS(255,223)编码/解码电路和3万门卫星上行信道BCH(64,56)编码/解码电路,这几个项目已先后通过有关单位的验收。1999年~2000年期间,我们也为自己的科研项目——小波(Wavelet)图像压缩,成功地设计了小波卷积器和改进的零修剪树算法(即SPIHT算法)的RTL级Verilog HDL模型,不但成功地对该模型进行了仿真和综合,而且制成的可重新配置硬线逻辑(采用ALTERA FLEX10K系列CPLD/10/30/50各一片)的PCI线路板,能完成约2000条C语句的程序才能完成的图像/解压缩算法。运算结果与软件完成的完全一致,而且速度比用Pentium II 333 MHz CPU的PC机更快,而PCI线路板上基本时钟仅为8.33 MHz。近年来由我协助指导的JPEG 2000算法硬线逻辑的前端设计,在清华同行的努力下已完成FPGA验证并开始进入投片验证阶段。可见这种新的设计方法的潜力。

本书是在1998年由北京航空航天大学出版社出版的《复杂数字电路与系统的Verilog HDL设计技术》和高教出版社出版的《从算法设计到硬线逻辑的实现》和《实验练习与Verilog语法手册》基础上全面修改和扩充而成,是一本既有理论又有实践的设计大全。由于教学、科研和实验室的各项工作很忙,我只能利用零碎时间,一点一点地把积累的教学经验和新收集到的材料补充输入到计算机中并逐步加以整理。因为我们使用Verilog HDL设计复杂数字逻辑电路总共也只有近10年的时间,水平并不高,书中谬误之处在所难免,敬请读者及时把意见反馈给我。我之所以在原出版基础上把这本书改版推出,是想把我们在采用Verilog HDL设计方法上新积累的一些经验与读者分享,把原教材中一些不足的地方作一补充和修改,在大学生和研究生中加快Ver-

ilog HDL 设计技术的推广,尽快培养一批掌握先进设计技术的跨世纪的人才。期望本书能在这过程中起到抛砖引玉的作用。

回想起来,这本书实质上是我们实验室全体老师和同学们的劳动成果,其中在 EDA 实验室工作过的历届研究生和本科生张琰、山岗、王静璇、田玉文、冯文楠、杨柳、傅红军、龚剑、王书龙和胡瑛等都帮我做了许多工作,如部分素材的翻译、整理、录入和一些 Verilog HDL 模块的设计和验证。而我做的工作只是收集了全书的素材、翻译和理解素材中一些较难的概念并结合教学经验把它们编写成通顺的段落,以及全书文稿最后的组织、整理和补充,使其达到能够出版的要求。实验室的董金明和杨惠军老师也给了我许多帮助和鼓励。特别是董金明老师一直以他努力工作的实际行动给我以最有力的鼓励和鞭策,使我不能懈怠。在本书出版之际,我衷心地感谢在编写本书过程中所有给过我帮助和鼓励的老师和同学们。

编 者

2003 年 1 月 30 日

于北京航空航天大学逸夫科学馆 EDA 实验室

绪 论

我们知道构成数字逻辑系统的基本单元是与门、或门和非门,它们都是由三极管、二极管和电阻等器件构成,并能执行相应的开关逻辑操作;与门、或门和非门又可以构成各种触发器,实现状态记忆。在数字电路基础课程中,在了解这些逻辑门和触发器的构成和原理后,把它们作为抽象的理想器件来考虑,学习如何用布尔代数和卡诺图简化方法来设计一些简单的组合逻辑电路和时序电路。这些基础知识从理论上了解一个复杂的数字系统,例如 CPU 等都可以由这些基本单元组成。但真正如何来设计一个极其复杂的数字系统,如何验证设计的逻辑系统功能是否正确,过去一直缺少更好的手段和方法。本教程就是讲解如何利用 Verilog 硬件描述语言来设计和验证这样一个复杂的数字系统的方法。下面就复杂数字系统的概念、用途和几个有关的基本问题做一些说明。

1. 为什么要设计专用的复杂数字系统

现代计算机与通信系统的电子设备中广泛使用了数字信号处理专用集成电路,它们主要用于数字信号传输中所必需的滤波、变换、加密、解密、编码、解码、纠错、压缩和解压缩等操作。这些操作从本质上说都是数学运算;而从原则上讲,又完全可以用计算机或微处理器来完成。这就是为什么常用 C、Pascal 或汇编语言来编写程序,以研究算法的合理性和有效性的道理。

在数字信号处理的领域内有相当大的一部分工作是可以事后处理的,即利用通用的计算机系统来处理这类问题。如在石油地质调查中,通过钻探和一系列的爆破,记录各种地层的回波数据,然后去除噪声等无用信息,并用计算机对这些数据进行处理,最后得到地层的构造,从而找到埋藏的石油。因为地层不会在几年内有明显的变化,因此花数十天乃至更长的时间把地层的构造分析清楚也能满足要求。这种类型的数字信号处理是非实时的,在通用的计算机上通过编写、修改和运行程序,分析程序运行的结果就能满足需要。

还有一类数字信号处理必须在规定的时间内完成,例如在军用无线通信系统和机载雷达系统中常常需要对检测到的微弱信号进行增强、加密、编码、压缩,而在接收端必须及时地解压缩、解码和解密并重现清晰的信号。很难想像用一个通用的计算机系统来完成这项工作。因此,我们不得不自行设计非常轻便而小巧的高速专用硬件系统来完成该任务。

有的数字信号处理对时间的要求非常苛刻,以至于用高速的通用微处理器芯片也无法在规定的时间内完成必要的运算。因此,必须为这样的运算设计一个专用的高速硬线逻辑电路,在高速 FPGA 器件上实现或制成高速专用集成电路。这是因为通用微处理器芯片是为一般目的而设计的,运算的步骤必须通过程序编译后生成的机器码指令加载到存储器中,然后在微处理器芯片控制下,按时钟的节拍,逐条取出指令、分析指令和执行指令,直至程序的结束。微处理器芯片中的内部总线和运算部件也是为通用目的而设计,即使是专为信号处理而设计的通用微处理器,因为它的通用性,也不可能为某一个特殊的算法来设计一系列的专用的运算电路,而且其内部总线的宽度也不能随意改变,只有通过改变程序,才能实现这个特殊的算法,因而其运算速度也受到限制。

本教程的目的是想通过对数字信号处理、计算(Computing)、算法和数据结构、编程语言和程序、体系结构和硬线逻辑等基本概念的介绍,了解算法与硬线逻辑之间的关系,从而引入利用 Verilog HDL 硬件描述语言设计复杂的数字逻辑系统的概念和方法。向读者展示一种 20 世纪 90 年代才真正开始在美国等先进的工业国家逐步推广的数字逻辑系统的设计方法,借助于这种方法,在电路设计自动化仿真和综合工具的帮助下,只要对并行计算微体系结构有一定程度的了解,对有关算法有深入的研究,我们完全有能力设计并制造出具有自己知识产权的 DSP(数字信号处理)类和任何复杂的数字逻辑集成电路芯片,为我国的电子工业和国防现代化做出应有的贡献。

2. 数字信号处理

大规模集成电路设计制造技术和数字信号处理技术,近 30 年来,各自得到了迅速的发展。这两个表面上看来没有什么关系的技术领域实质上是紧密相关的。因为数字信号处理系统往往要进行一些复杂的数学运算和数据处理,并且又有实时响应的要求,它们通常是由高速专用数字逻辑系统或专用数字信号处理器所构成,电路是相当复杂的。因此只有在高速大规模集成电路设计制造技术进步的基础上,才有可能实现真正有意义的实时数字信号处理系统。对实时数字信号处理系统的要求不断提高,也推动了高速大规模集成电路设计制造技术的进步。现代专用集成电路的设计是借助于电子电路设计自动化(EDA)工具完成的。学习和掌握硬件描述语言(HDL)是使用电子电路设计自动化(EDA)工具的基础。

3. 计算(Computing)

说到数字信号处理,自然会想到数学计算(或数学运算)。现代计算机和通信系统中广泛采用了数字信号处理的技术和方法。其基本思路是先把信号用一系列的数字来表示,如连续的模拟信号,则通过采样和从模拟量到数字量的转换,把信号转换成一系列的数字信号,然后对这些数字信号进行各种快速的数学运算。其目的是多种多样的,有的是为了加密;有的是通过编码来减少误码率以提高信道的通信质量;有的是为了去掉噪声等无关的信息,称为滤波;有的是为了数据的压缩以减少占用的频道等。有时也把某些种类的数字信号处理运算称为变换,如离散傅里叶变换(DFT)、离散余弦变换(DCT)和小波变换(Wavelet T)等。

这里所说的计算是从英语 Computing 翻译过来的,它的含义要比单纯的数学计算广泛得多。“Computing 这门学问研究怎样系统地有步骤地描述和转换信息,实质上是一门覆盖了多个知识和技术范畴的学问,其中包括了计算的理论、分析、设计、效率和应用。它提出的最基本的问题是什么样的工作能自动完成,什么样的不能”(摘自 Denning et al, “Computing as a Discipline,” Communication of ACM, January, 1989)。

本书中凡提到“计算”这个词处,指的就是上面一段中 Computing 所包含的意思。由传统的观点出发,可以从三个不同的方面来研究计算,即从数学、科学和工程的不同角度;由比较现代的观点出发,可以从四个主要的方面来研究计算,即从算法和数据结构、编程语言、体系结构、软件和硬件设计方法学。本绪论的目的是想让读者对设计复杂数字系统有一个全面的了解,从而加深对掌握 Verilog HDL 设计方法必要性的认识。一个复杂的数字系统设计往往是一个从算法到由硬线连接的门级逻辑结构,再映射到硅片的逐步实现的过程。因此,我们将从算法和数据结构、编程语言和程序、微体系结构和硬线逻辑以及设计方法学等方面的基本概念出发来研究和探讨用于数字信号处理等领域的复杂硬线逻辑电路的设计技术和方法。特别强调利用 Verilog 硬件描述语言的 Top-Down 设计方法的介绍。

4. 算法和数据结构

为了准确地表示特定问题的信息并顺利地解决有关的计算问题,需要采用一些特殊方法并建立相应的模型。所谓算法就是解决特定问题的有序步骤,所谓数据结构就是解决特定问题的相应的模型。

5. 编程语言和程序

程序员利用一种由专家设计的既可以被人理解,也可以被计算机解释的语言来表示算法问题的求解过程。这种语言就是编程语言,由它所表达的算法问题的求解过程就是程序。而 C、Pascal、Fortran、Basic 语言或汇编语言是几种常用的编程语言。如果只研究算法,只在通用的计算机上运行程序或利用通用的 CPU 来设计专用的微处理器嵌入系统,掌握上述语言就足够了。如果还需要设计和制造能进行快速计算的硬线逻辑专用电路,必须学习数字电路的基本知识和硬件描述语言。因为现代复杂数字逻辑系统的设计都是借助于 EDA 工具完成的,无论电路系统的仿真和综合都需要掌握硬件描述语言。在本书中将要比较详细地介绍 Verilog 硬件描述语言。

6. 系统的微体系结构和硬线连接的门级逻辑

计算电路究竟是如何构成的?为什么它能有效地和正确地执行每一步程序?它能不能用另外一种结构方案来构成?运算速度还能不能再提高?所谓计算微体系结构就是回答以上问题并从硬线逻辑和软件两个角度一起来探讨某种结构的计算机的性能潜力。比如, Von Neumann(冯·诺依曼)于 1945 年设计的 EDVAC 电子计算机,它的结构是一种最早的顺序机,该机执行标量数据的计算机系统结构。顺序机是从位串行操作到字并行操作,从定点运算到浮点运算逐步改进过来的。由于 Von Neumann 系统结构的程序是顺序执行的,所以速度很慢。随着硬件技术的进步,不断有新的计算机体系结构产生,其计算性能也在不断提高。计算机体系结构是一门讨论和研究通用计算机的中央处理器如何提高运算速度性能的学问。对计算机中央处理器微体系结构的了解是设计高性能的专用的硬线逻辑系统的基础,因此本书将通过一个简化的 RISC_CPU 的设计实例对系统结构的基本概念加以初步的介绍。但由于本书的重点是利用 Verilog HDL 进行复杂数字电路的设计技术和方法,大量的篇幅将介绍利用 HDL 进行设计的步骤、语法要点、可综合的风格要点、同步有限状态机和由浅入深的设计实例。至于有关处理器微体系结构的深入了解和高速标量计算逻辑的微结构等专门知识和设计诀窍,将在以后推出的新书中介绍。

7. 设计方法学

复杂数字系统的设计是一个把思想(即算法)转化为实际数字逻辑电路的过程。我们知道,同一个算法可以用不同结构的数字逻辑电路来实现,这从运算的结果来说可能是完全一致的,但其运算速度和性能价格比可以有很大的差别。我们可用许多种不同的方案来实现实时完成算法运算的复杂数字系统电路,下面列出了常用的四种方案:

第一种,以专用微处理机芯片为中心来完成算法所需的电路系统;

第二种,用高密度的 FPGA(从几万门到几百万门);

第三种,设计专用的大规模集成电路(ASIC);

第四种,利用现成的微处理机的 IP 核并结合专门设计的高速 ASIC 运算电路。

究竟采用什么方案要根据具体项目的技术指标、经费、时间进度和批量综合考虑而定。

在上述第二、第三、第四种设计方案中,电路结构的考虑和决策至关重要。有的电路结构

速度快,但所需的逻辑单元多,成本高;而有的电路结构速度慢,但所需的逻辑单元少,成本低。复杂数字逻辑系统设计的过程往往需要通过多次仿真,从不同的结构方案中找到一种符合工程性能价格比最好的结构。一个优秀的有经验的设计师,能通过硬件描述语言的顶层仿真较快地确定合理的系统电路结构,减少由于总体结构设计不合理而造成的返工,从而大大加快系统的设计过程。

8. 专用硬线逻辑与微处理器的比较

在信号处理专用计算电路的设计中,以专用微处理器芯片为中心来构成完成算法所需的电路系统是一种较好的办法。可以利用现成的微处理器开发系统,在算法已用 C 语言验证的基础上,在开发系统工具的帮助下,将该 C 语言程序转换为专用微处理器的汇编,然后再编译为机器代码,最后加载到样机系统的存储区,即可以在开发系统工具的环境下开始相关算法的运算仿真或运算。采用这种方法,设计周期短、可以利用的资源多,但速度、能耗、体积等性能受该微处理器芯片和外围电路的限制。

用高密度的 FPGA(从几万门到几百万门)来构成完成算法所需的电路系统也是一种较好的办法。我们必须购置有关的 FPGA 开发环境、布局布线和编程工具。有些 FPGA 厂商提供的开发环境不够理想,其仿真工具和综合工具性能不够好,还需要利用性能较好的硬件描述语言仿真器、综合工具,才能有效地进行复杂的 DSP 硬线逻辑系统的设计。由于 FPGA 是一种通用的器件,它的基本结构决定了只对某一种特殊的应用,其性能不如专用的 ASIC 电路。

采用自行设计的专用 ASIC 系统芯片(System On Chip),即利用现成的微处理机 IP 核或根据某一特殊应用设计的微处理机核(也可以没有通用的微处理机核),并结合专门设计的高速 ASIC 运算电路,能设计出性能价格比最高的理想数字信号处理系统。这种方法结合了微处理器和专用的大规模集成电路的优点。由于微处理器 IP 核的挑选结合了算法和应用的特点,又加上专用的 ASIC 在需要高速部分的增强,能“量体裁衣”,因而各方面性能优越。但由于设计和制造周期长、投片成本高,往往只有经费充足、批量大的项目或重要的项目才采用这一途径。当然性能优良的硬件描述语言仿真器、综合工具是不可缺少的,另外对所采用的半导体厂家基本器件库和 IP 库的深入了解也是必须的。

以上所述算法的专用硬线逻辑的实现都需要对算法有深入的了解,还需掌握硬件描述语言和相关的 EDA 仿真、综合和布局布线工具。

9. C 语言、Matlab 与硬件描述语言在算法运算电路设计的关系和作用

数字电路设计工程师一般都学习过编程语言、数字逻辑基础、各种 EDA 软件工具的使用。就编程语言而言,国内外大多数学校都以 C 语言为标准,目前很少有学校使用 Pascal 和 Fortran;而 Matlab 则是一个常用的数学计算软件包,有许多现成的数学函数可以利用,大大节省了复杂函数的编程时间,Matlab 也提供手段可以与 C 程序模块方便地接口,因此用 Matlab 来做数学计算系统的行为仿真常常比直接用 C 语言方便,能很快生成有用的数据文件和表格,直接用于算法正确性的验证。

基础算法的描述和验证常用 C 语言来做。例如要设计 Reed-Solomen 编码/解码器,必须先深入了解 Reed-Solomen 编码/解码的算法,再编写 C 语言的程序来验证算法的正确性。运行描述编码器的 C 语言程序,把在数据文件中的多组待编码的数据转换为相应的编码后将数据并存入文件。再编写一个加干扰用的 C 语言程序,用于模拟信道。它能产生随机误码位(并把误码位个数控制在纠错能力范围内)将其加入编码后的数据文件中。运行该加扰程序,

产生带误码位的编码后的数据文件;然后再编写一个解码器的 C 语言程序,运行该程序把带误码位的编码文件解码为另一个数据文件。只要比较原始数据文件和生成的文件便可知道编码和解码的程序是否正确(能否自动纠正纠错能力范围内的错码位)。用这种方法就可以来验证算法的正确性。但这样的数据处理其运行速度只与程序的大小和计算机的运行速度有关,也不能独立于计算机而存在。如果要设计一个专门的电路来进行这种对速度有要求的实时数据处理,除了以上介绍的 C 程序外,还须编写硬件描述语言(如 Verilog HDL 或 VHDL)程序,进行仿真以便从电路结构上保证算法能在规定的时间内完成,并能通过与前端和后端的设备接口正确无误地交换(输入/输出)数据。

用硬件描述语言(HDL)的程序设计硬件的好处在于易于理解、易于维护、调试电路速度快、有许多的易于掌握的仿真、综合和布局布线工具,还可以用 C 语言配合 HDL 来做逻辑设计的布线前和布线后仿真,验证功能是否正确。

在算法硬件电路的研制过程中,计算电路的结构和芯片的工艺对运行速度有很大的影响。所以在电路结构完全确定之前,必须经过多次仿真:

- 1) C 语言的功能仿真;
- 2) C 语言的并行结构仿真;
- 3) Verilog HDL 的行为仿真;
- 4) Verilog HDL RTL 级仿真;
- 5) 综合后门级结构仿真;
- 6) 布局布线后仿真;
- 7) 电路实现验证。

下面介绍用 C 语言配合 Verilog HDL 来设计算法的硬件电路块时考虑的三个主要问题:

- 为什么选择 C 语言与 Verilog HDL 配合使用;
- C 语言与 Verilog HDL 的使用有何限制;
- 如何利用 C 语言来加速硬件的设计和故障检测。

(1) 为什么选择 C 语言与 Verilog 配合使用:首先,C 语言很灵活,查错功能强,还可以通过 PLI(编程语言接口)编写自己的系统任务直接与硬件仿真器(如 Verilog-XL)结合使用。C 语言是目前世界上应用最为广泛的一种编程语言,因而 C 程序的设计环境比 Verilog HDL 更完整。此外,C 语言可应用于许多领域,有可靠的编译环境,语法完备,缺陷较少。比较起来,Verilog 语言只是针对硬件描述的,在别处使用(如用于算法表达等)并不方便。而且 Verilog 的仿真、综合、查错工具等大部分软件都是商业软件,与 C 语言相比缺乏长期大量的使用,可靠性较差,亦有很多缺陷。所以,只有在 C 语言的配合使用下,Verilog 才能更好地发挥作用。

面对上述问题,最好的方法是 C 语言与 Verilog 语言相辅相成,互相配合使用。这就是既要利用 C 语言的完整性,又要结合 Verilog 对硬件描述的精确性,来更快更好地设计出符合性能要求的硬件电路系统。利用 C 语言完善的查错和编译环境,设计者可以先设计出一个功能正确的设计单元,以此作为设计比较的标准。然后,把 C 程序一段一段地改写成用并型结构(类似于 Verilog)描述的 C 程序,此时还是在 C 的环境里,使用的依然是 C 语言。如果运行结果都正确,就将 C 语言关键字用 Verilog 相应的关键字替换,进入 Verilog 的环境。将测试输入同时加到 C 与 Verilog 两个单元,将其输出做比较。这样很容易发现问题的所在,然后更

正,再做测试,直至正确无误。剩下的工作就交给后面的设计工程师继续做。

(2) C 语言与 Verilog 语言互相转换中存在的问题是,混合语言设计流程往往会在两种语言的转换中遇到许多难题。例如,怎样把 C 程序转换成类似 Verilog 结构的 C 程序,来增加并行度,以保证用硬件实现时运行速度达到设计要求;又如怎样不使用 C 中较抽象的语法:例如迭代,指针,不确定次数的循环等,也能来表示算法,因为转换的目的是要用可综合的 Verilog 语句来代替 C 程序中的语句,而可用于综合的 Verilog 语法是相当有限的,往往找不到相应的关键字来替换。

C 程序是一行接一行依次执行的,属于顺序结构,而 Verilog 描述的硬件是可以在同一时间同时运行的,属于并行结构,这两者之间有很大的冲突。而 Verilog 的仿真软件也是顺序执行的,在时间关系上同实际的硬件是有差异的,可能会出现一些无法发现的问题。

Verilog 可用的输出输入函数很少,C 语言的花样则很多,转换过程中会遇到一些困难。C 语言的函数调用与 Verilog 中模块的调用也有区别。C 程序调用函数是没有延时特性的,一个函数是惟一确定的,对同一个函数的不同调用是一样的。而 Verilog 中对模块的不同调用是不同的,即使调用的是同一个模块,必须用不同的名字来指定。Verilog 的语法规则很死,限制很多,能用的判断语句有限。仿真速度较慢,查错功能差,错误信息不完整。仿真软件通常也很昂贵,而且不一定可靠。C 语言没有时间关系,转换后的 Verilog 程序必须做到没有任何外加的人工延时信号,也就是必须表达为有限状态机,即 RTL 级的 Verilog,否则将无法使用综合工具把 Verilog 源代码转化为门级逻辑。

(3) 如何利用 C 语言来加快硬件的设计和查错:下表中列出了常用的 C 语言与 Verilog 相对应的关键字与控制结构。

C 语言	Verilog 语言
sub-function	module, function, task
if-then-else	if-then-else
Case	Case
{,}	begin, end
For	For
While	While
Break	Disable
Define	Define
Int	Int
Printf	monitor, display, strobe

下表中,列出了 C 语言与 Verilog 相对应的运算符。

C 语言	Verilog 语言	功 能
*	*	乘
/	/	除
+	+	加
-	-	减
%	%	取模
!	!	反逻辑
&&	&&	逻辑与
		逻辑或
>	>	大于
<	<	小于
>=	>=	大于等于
<=	<=	小于等于
==	==	等于
!=	!=	不等于
~	~	位反相
&	&	按位逻辑与
		按位逻辑或
^	^	按位逻辑异或
~^	~^	按位逻辑同或
>>	>>	右移
<<	<<	左移
?:	?:	同等于 if-else 叙述

从上面的讨论可以总结如下：

- C 语言与 Verilog 硬件描述语言可以配合使用，辅助设计硬件。
- C 语言与 Verilog 硬件描述语言类似，只要稍加限制，C 语言的程序很容易转成 Verilog 的行为程序。

美国和中国台湾地区逻辑电路设计和制造厂家大都以 Verilog HDL 为主，中国大陆地区目前学习使用 Verilog HDL 逐渐超过 VHDL。到底选用 VHDL 或是 Verilog HDL 来配合 C 一起用，就留给各位同学自行去决定。但从学习的角度来看，Verilog HDL 比较简单，也与 C 语言较接近，容易掌握。从使用的角度看，支持 Verilog 硬件描述语言的半导体厂家也较支持 VHDL 的多。

总 结

在绪论中我们全面介绍了信号处理与硬线逻辑设计的关系,以及有关的基本概念。引入了 Verilog HDL 硬件描述语言,向读者展示一种 20 世纪 90 年代才真正开始在美国等先进的工业国家逐步推广的数字逻辑系统的设计方法。在具有必要的基础知识的前提下,借助于这种方法,在电路设计自动化仿真和综合工具的支持下,我们完全有能力设计并制造出有自己知识产权的 DSP(数字信号处理)类和任何复杂的数字逻辑集成电路芯片,为我国的电子工业和国防现代化作出应有的贡献。在下面的各章里我们将分步骤地详细介绍这种设计方法。

思考题

1. 什么是信号处理电路?
2. 为什么要设计专用的信号处理电路?
3. 什么是实时处理系统?
4. 为什么要用硬件描述语言来设计复杂的算法逻辑电路?
5. 能不能完全用 C 语言来代替硬件描述语言进行算法逻辑电路的设计?
6. 为什么在算法逻辑电路的设计中需要用 C 语言和硬件描述语言配合使用来提高设计效率?

出版者的话

利用晶体管等分立元件在面包板上搭建一个与门、触发器这样一个刀耕火种时代已一去不复返了,而利用硬件描述语言在计算机上编制、调试一个几万门乃至几百万门以上的复杂数字电路的时代已经来临。美国及工业发达国家用 Verilog HDL 语言早在 10 年前如火如荼进行。全美 200 多所大学,10 余万工程人员讲授并实践着用硬件描述语言来设计复杂数字系统。中国在此领域差距甚大。

作者 60 年代毕业于清华大学自控系计算与技术装置专业,近 40 年来一直从事嵌入式数字系统的设计研究工作,曾作为访问学者在国外研究多年,深感这一领域的落后与学校课程设置和教学条件有关,与人才缺乏有关。为此,于 1995 年受校方之委托,开始筹建国内首个 EDA 实践室,并利用世行贷款引入技术与设备,经过七八年的艰苦努力,对 HDL 语言及其设计技术进行了深入研究,至使不少学生走向社会,科技成果也用于实践,国内首本 Verilog HDL 原理及设计技术在北航出版社出版。图书的出版影响了一批批读者,促进了 IC 工业的发展。

《Verilog 数字系统设计教程》的出版,集作者多年的教学和科研经验,并在以往出版的基础上,不断提高与精炼,突破传统的章节风格,采用“讲授”方式,灵活自如地将知识传授给每一个读者。

我们认为:随着祖国经济建设的不断发展,信息、通信等行业都与高校教学密切相关,掌握和运用 HDL 技术犹如使用万用表一样,将变得如此普遍和重要,作为高校有责任将这一技术作为必修课固定下来,推广出去,以此来提高我们的水平,缩短与人家的差距。

我们将此书推荐给大家,希望读者钟爱。

2003 年 6 月 1 日

目 录

绪 论

第一部分 初 级 篇

第一讲 Verilog 的基本知识

1.1 硬件描述语言 HDL	1
1.2 Verilog HDL 的历史	2
1.2.1 什么是 Verilog HDL	2
1.2.2 Verilog HDL 的产生及发展	2
1.3 Verilog HDL 和 VHDL 的比较	3
1.4 Verilog 目前的应用情况和适用的设计	4
1.5 采用 Verilog HDL 设计复杂数字电路的优点	5
1.5.1 传统设计方法——电路原理图输入法	5
1.5.2 Verilog HDL 设计法与传统的电路原理图输入法的比较	5
1.5.3 Verilog 的标准化与软核的重用	5
1.5.4 软核、固核和硬核的概念以及它们的重用	6
1.6 Verilog HDL 的设计流程简介	6
1.6.1 自顶向下(Top-Down)设计的基本概念	6
1.6.2 层次管理的基本概念	7
1.6.3 具体模块的设计编译和仿真的过程	7
1.6.4 对应具体工艺器件的优化、映象和布局布线	7
1.7 小 结	8
思考题	9

第二讲 Verilog 语法的基本概念

概 述	10
2.1 Verilog 模块的基本概念	11
2.2 Verilog 用于模块的测试	13
2.3 小 结	15
思考题	15

第三讲 常用 Verilog 语法之一

- 概 述 17
- 3.1 模块的结构..... 17
 - 3.1.1 模块的端口定义..... 17
 - 3.1.2 模块内容..... 18
 - 3.1.3 理解要点..... 19
 - 3.1.4 要点总结..... 19
- 3.2 数据类型及其常量及变量..... 20
 - 3.2.1 常 量..... 20
 - 3.2.2 变 量..... 23
- 3.3 运算符及表达式..... 25
 - 3.3.1 基本的算术运算符..... 26
 - 3.3.2 位运算符..... 26
- 3.4 小 结..... 28
- 思考题 28

第四讲 常用 Verilog 语法之二

- 概 述 29
- 4.1 逻辑运算符..... 29
- 4.2 关系运算符..... 30
- 4.3 等式运算符..... 30
- 4.4 移位运算符..... 31
- 4.5 位拼接运算符..... 31
- 4.6 缩减运算符..... 32
- 4.7 优先级别..... 32
- 4.8 关 键 词 33
- 4.9 赋值语句和块语句..... 33
 - 4.9.1 赋值语句..... 33
 - 4.9.2 块语句..... 35
- 4.10 小 结 38
- 思考题 38

第五讲 常用 Verilog 语法之三

- 概 述 39
- 5.1 条件语句..... 39
 - 5.1.1 if_else 语句..... 39
 - 5.1.2 case 语句..... 42
 - 5.1.3 使用条件语句不当在设计中生成了原本没想到有的锁存器..... 45
- 5.2 循环语句..... 46

5.2.1 forever 语句	46
5.2.2 repeat 语句	46
5.2.3 while 语句	47
5.2.4 for 语句	47
5.3 小 结	49
思考题	49
第六讲 常用 Verilog 语法之四	
概 述	50
6.1 结构说明语句	50
6.1.1 initial 语句	50
6.1.2 always 语句	51
6.2 task 和 function 说明语句	52
6.3 系统函数和任务	56
6.4 小 结	60
思考题	60
第七讲 常用 Verilog 语法之五	
概 述	61
7.1 系统任务 \$monitor	61
7.2 时间度量系统函数 \$time	62
7.3 系统任务 \$finish	63
7.4 系统任务 \$stop	63
7.5 系统任务 \$readmemb 和 \$readmemh	64
7.6 系统任务 \$random	65
7.7 编译预处理	66
7.7.1 宏定义 `define	66
7.7.2 “文件包含”处理 `include	68
7.7.3 时间尺度 `timescale	71
7.7.4 条件编译命令 `ifdef、`else、`endif	73
7.8 小 结	73
思考题	74
第八讲 常用 Verilog 语法总结	
概 述	75
总 结	86