高等学校教材

从算法设计到硬线逻辑的实现

——复杂数字逻辑系统的 Verilog HDL 设计技术和方法

夏宇闻　编著

高等教育出版社

内容提要

本书着重介绍20世纪90年代才开始在美国等先进的工业国家逐步推广的用硬件描述语言(Verilog HDL)建模、仿真和综合的设计方法和技术。本书从算法和计算的基本概念出发,讲述把复杂算法逐步分解成简单的操作步骤,最后由硬线逻辑电路系统来实现该算法的技术和方法。这种硬线逻辑电路系统就是广泛应用于各种现代通信电子设备与计算机系统中的专用集成电路(ASIC)或FPGA。主要内容包括:基本概念、Verilog HDL的基本语法、不同抽象级别的Verilog HDL模型以及有限状态机和可综合风格的Verilog HDL实例等。本书可作为计算机或电子类本科高年级或研究生的教材,也可供在数字系统设计领域工作的工程师参考或作为自学教材。本书配有《从算法设计到硬线逻辑的实现——实验练习与Verilog语法手册》出版。

图书在版编目(CIP)数据

从算法设计到硬线逻辑的实现:复杂数字逻辑系统的Verilog HDL设计技术和方法/夏宇闻编著.—北京:高等教育出版社,2001

ISBN 7-04-009252-2

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

中国版本图书馆CIP数据核字(2001)第02980号

责任编辑 董建波 **封面设计** 李卫青 **版式设计** 马静如

责任校对 王效珍 **责任印制** 张小强

从算法设计到硬线逻辑的实现——复杂数字逻辑系统的Verilog HDL设计技术和方法

夏宇闻 编著

出版发行	高等教育出版社		
社 址	北京市东城区沙滩后街55号	邮政编码	100009
电 话	010-64054588	传 真	010-64014048
网 址	http://www.hep.edu.cn http://www.hep.com.cn		
经 销	新华书店北京发行所		
排 版	高等教育出版社照排中心		
印 刷	北京市鑫鑫印刷厂		
开 本	787×1092 1/16	版 次	2001年2月第1版
印 张	18.75	印 次	2001年2月第1次印刷
字 数	450 000	定 价	22.00元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

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。我们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。作为一名在大学讲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。通过几年的摸索、实践,我们掌握了利用Verilog HDL设计复杂数字电路的仿真和综合技术。在此基础上我们为航天部等有关单位设计了13万门卫星信道加密用复杂数字电路,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。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)的硬线逻辑的Verilog HDL模型,不但成功地进行了仿真和综合,而且制成的可重配置硬线逻辑(采用ALTERA FLEX10K系列CPLD/10/30/50各一片)的PCI线路板,能完成约2 000条C语句的程序才能完成的图像压缩/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 333MHz CPU的PC机更快,而PCI线路板上基本时钟仅为8.33MHz。可见这种新的设计方法的潜力。

本书是对1998年由北京航空航天大学出版社出版的教材基础上进行全面的修改扩充而成的。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。到现在又两年过去了,新版书较之最初的版本总算有了一个初步的样子。我把书名改为《从算法设计到硬线逻辑的实现——复杂数字电路与系统的Verilog HDL设计技术》,这是因为本书是围绕着算法的实现来介绍Verilog HDL设计方法的。因为我们使用Verilog HDL设计复杂数字逻辑电路总共也只有5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。我之所以匆匆把这本书改版推出,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些补充和修改。在大学生和研究生中加快Verilog 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。期望本书能在这一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、龚剑、傅红军、王书龙和胡英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

2000 年 8 月 30 日

于北京航空航天大学逸夫科学馆 EDA 实验室

目　录

第一章　数字信号处理、计算、程序、算法和硬线逻辑设计的基本概念

引　言

现代的计算机与通信设备中，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、纠检错、压缩、解压缩等操作。这些处理工作从本质上说都是数学运算。从原则上讲，它们完全可以用计算机或微处理器来完成。这就是为什么我们常用 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。可以利用通用的计算机系统来处理这类问题。如在石油地质勘探中，通过钻探和一系列的爆破，记录下各种地层的回波数据，然后用计算机对这些数据进行处理，去除噪声等无用信息，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，因此花几十天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，用通用的计算机就能满足需要。

还有一类数字信号处理必须在规定的时间内完成，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增强、加密、编码、压缩，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，不得不自行设计非常轻便小巧的高速专用硬件系统来完成该任务。

有的数字信号处理则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须的运算。因此必须为这样的运算设计专用的硬线逻辑电路，这可以在高速 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，然后执行指令，直至程序结束。微处理器芯片中的内部总线和运算部件也是为通用的目的而设计，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。因而其运算速度就受到限制。

本章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，只要对并行的计算结构有一定程度的了解，对有关算法有深入的研究，就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。

1.1　数字信号处理

近30年来，大规模集成电路设计制造技术和数字信号处理技术，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的处理，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。

1.2　计算(Computing)

说到数字信号处理，自然就会想到数学计算(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。基本思路是先把信号用一系列的数字来表示，如果是连续的模拟信号，则需通过采样和模/数转换，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，这样做的目的是多种多样的，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也可以称为滤波，有的是为了数据的压缩以减少占用的频道……。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)、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，实质上它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。它提出的基本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, 1989)。

本文中凡提到计算这个词，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。本书的主题是从算法到硬线逻辑的实现，因此将从算法和数据结构、编程语言和程序、体系结构和硬线逻辑以及设计方法学等基本概念出发，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。重点介绍利用 Verilog 硬件描述语言的自顶向下(Top-Down)设计方法。

1.3　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，所谓数据结构就是描述特定问题的

相应模型。

1.4 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。我们已经熟悉通过编写程序来解决计算问题，如 C、PASCAL、FORTRAN、BASIC 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，就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。在本书中将比较详细地介绍 Verilog 硬件描述语言。

1.5 系统结构和硬线逻辑

计算机究竟是如何构成的？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？所谓计算机系统结构就是回答以上问题的，并从硬线逻辑和软件两个角度一起来探讨某种结构计算机的性能潜力。比如，Von Neumann(冯·诺依曼)在 1945 年，设计的 EDVAC 电子计算机，它的结构是一种最早的顺序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，不断有新的计算机系统结构产生，其计算性能也在不断提高。计算机系统结构是一门讨论和研究通用的计算机中央处理器如何提高运算速度和性能的学问。对计算机系统结构的深入了解是设计高性能的专用硬线逻辑系统的基础，因此也是本书讨论的内容之一。但由于本书重点介绍的是利用 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 Verilog 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。

1.6 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。我们都知道，同一个算法可以用不同结构的数字逻辑电路来实现，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则有很大的差别。我们可用许多种不同的方案来实现能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：1) 以专用微处理机芯片为中心来构成完成算法所需的电路系统；2) 用高密度的 FPGA(从几万门到百万门)；3) 设计专用的大规模集成电路(ASIC)；4) 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、时间进度和生产批量综合考虑而定。

在上述第 2)、第 3)、第 4)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技术要求的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进程。

1.7 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,在系统开发工具的帮助下,把该 C 语言程序转换为专用微处理器的汇编语言程序再编译为机器代码,然后加载到样机系统的存储区,即可以在系统开发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(从几万门到几十万门)来构成算法所需的电路系统也是一种较好的办法。但必须购置有关的 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,它的基本结构必须适用于多种功能的电路,因而对某一种特殊应用性能不如专门为特殊应用设计的 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器 IP 核或根据某一特殊应用设计的微处理器核(也可以没有微处理器核),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在需要高速部分时的增强,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。

以上所述对算法的专用硬线逻辑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。

1.8 C 语言与硬件描述语言在算法运算电路设计中的关系和作用

数字电路设计工程师一般都学习过编程语言、数字逻辑基础和各种 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,只有少部分学校使用 PASCAL 和 FORTRAN。

算法的描述和验证常用 C 语言来完成。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),并将其加入编码后的数据文件中。运行该加干扰程序,产生带误码位编码的数据文件。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)。用这种方法就可以验证算法的正确性。但这样的数据处理其运行速度与程序的大小和计算机的运行速度有关,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,还需编写硬件描述语言(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,并能与前端和后端的设备或器件正确无误地交换数据。

用硬件描述语言(HDL)编写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、而且有许多易于掌握的仿真、综合和布局布线工具,还可以用C语言配合HDL来做逻辑设计的前后仿真,验证功能是否正确。

在研制实现算法的专用硬件电路过程中,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:

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的完整。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行结构(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

这样的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，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 可用的输入输出函数很少。C 语言的花样则很多，转换过程会遇到一些困难。

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，一个函数是唯一确定的，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，出错信息不完整。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 程序必须要能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

表 1.1 列出了常用的 C 与 Verilog 相对应的关键字与控制结构。

表　1.1

C	Verilog
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

表 1.2，列出了 C 与 Verilog 相对应的运算符。

表 1.2

C	Verilog	功 能
*	*	乘
/	/	除
+	+	加
-	-	减
%	%	取模
!	!	逻辑非
&&	&&	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
?:	?:	等同于 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 硬件描述语言很像，只要稍加限制，C语言程序很容易转成 Verilog 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 VHDL 的较多。但到底选用 VHDL 还是 Verilog HDL，就由读者去决定。但从学习的角度来看，我们认为 Verilog HDL 比较简单，也与 C 语言较接近，容易掌握。从使用的角度，支持 Verilog 硬件描述语言的半导体厂家也较支持 VHDL 的多。

小 结

本章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。简要介绍了 Verilog HDL 硬件描述语言以及 20 世纪 90 年代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，我们完全有能力设计并制造出有自主知识产权的 DSP(数字信号处理)类和任何复杂的数字逻辑集成电路芯片。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？

第二章　Verilog HDL 设计方法概述

引　　言

随着电子设计技术的飞速发展，专用集成电路(ASIC)和用户现场可编程门阵列(FPGA)的复杂度越来越高。数字通信、工业自动化控制等领域所用的数字电路及系统其复杂程度也越来越高，特别是需要设计具有实时处理能力的信号处理专用集成电路，并把整个电子系统综合到一个芯片上。设计并验证这样复杂的电路及系统已不再是简单的个人劳动，而需要综合许多专家的经验和知识才能够完成。由于电路制造工艺技术进步非常迅速，电路设计能力赶不上技术的进步。在数字逻辑设计领域，迫切需要一种共同的工业标准来统一对数字逻辑电路及系统的描述，这样就能把系统设计工作分解为逻辑设计(前端)和电路实现(后端)两个互相独立而又相关的部分。由于逻辑设计的相对独立性就可以把专家们设计的各种常用数字逻辑电路和系统部件(如 FFT 算法、DCT 算法部件)建成宏单元(Megcell)或软核(Soft - Core)库供设计者引用，以减少重复劳动，提高工作效率。电路的实现则可借助于综合工具和布局布线工具(与具体工艺技术有关)来自动地完成。

VHDL 和 Verilog HDL 这两种工业标准的产生顺应了历史的潮流，因而得到了迅速的发展。作为跨世纪的中国大学生应该尽早掌握这种新的设计方法，使我国在复杂数字电路及系统的设计竞争中逐步缩小与美国等先进的工业发达国家的差距。为我国下一个世纪的深亚微米百万门级的复杂数字逻辑电路及系统的设计培养一批技术骨干。

2.1　硬件描述语言 HDL

硬件描述语言(HDL——Hardware Description Language)是一种用形式化方法来描述数字电路和设计数字逻辑系统的语言。它可以使数字逻辑电路设计者利用这种语言来描述自己的设计思想，然后利用电子设计自动化(在下面简称为 EDA)工具进行仿真，再自动综合到门级电路，再用 ASIC 或 FPGA 实现其功能。目前，这种称之为高层次设计(High Level Design)的方法已被广泛采用。据统计，在美国硅谷目前约有 90% 以上的 ASIC 和 FPGA 已采用硬件描述语言进行设计。

硬件描述语言的发展至今已有 20 多年的历史，并成功地应用于设计的各个阶段：仿真、验证、综合等。到 20 世纪 80 年代时，已出现了上百种硬件描述语言，它们对设计自动化起到了极大的促进和推动作用。但是，这些语言一般各自面向特定的设计领域与层次，而且众多的语言使用户无所适从，因此急需一种面向设计的多领域、多层次、并得到普遍认同的标准硬件描述语言。进入 80 年代后期，硬件描述语言向着标准化的方向发展。最终，VHDL 和 Verilog HDL 语言适应了这种趋势的要求，先后成为 IEEE 标准。把硬件描述语言用于自动综合还只有短短的六、七年历史。最近三四年来，用综合工具把可综合风格的 HDL 模块自动转换为电路发展非常迅速，在美国已成为设计数字电路的主流。本书主要介绍如何来编写可综合风格的 Verilog HDL 模块，如

何借助于 Verilog 语言对所设计的复杂电路进行全面可靠的测试。

2.2　Verilog HDL 的历史

2.2.1　什么是 Verilog HDL

Verilog HDL 是硬件描述语言的一种，用于数字电子系统的设计。它允许设计者用它来进行各种级别的逻辑设计，可以用它进行数字逻辑系统的仿真验证、时序分析、逻辑综合。它是目前应用最广泛的一种硬件描述语言。据有关文献报道，目前在美国使用 Verilog HDL 进行设计的工程师大约有 60 000 人，全美国有 200 多所大学教授用 Verilog 硬件描述语言的设计方法。在我国台湾地区几乎所有著名大学的电子和计算机工程系都讲授 Verilog 有关的课程。

2.2.2　Verilog HDL 的产生及发展

Verilog HDL 是在 1983 年，由 GDA（GateWay Design Automation）公司的 Phil Moorby 首创的。Phil Moorby 后来成为 Verilog－XL 的主要设计者和 Cadence 公司（Cadence Design System）的第一个合伙人。在 1984—1985 年，Moorby 设计出了第一个关于 Verilog－XL 的仿真器，1986 年，他对 Verilog HDL 的发展又作出了另一个巨大贡献：即提出了用于快速门级仿真的 XL 算法。

随着 Verilog－XL 算法的成功，Verilog HDL 语言得到迅速发展。1989 年，Cadence 公司收购了 GDA 公司，Verilog HDL 语言成为 Cadence 公司的私有财产。1990 年，Cadence 公司决定公开 Verilog HDL 语言，于是成立了 OVI（Open Verilog International）组织来负责 Verilog HDL 语言的发展。基于 Verilog HDL 的优越性，IEEE 于 1995 年制定了 Verilog HDL 的 IEEE 标准，即 Verilog HDL1364

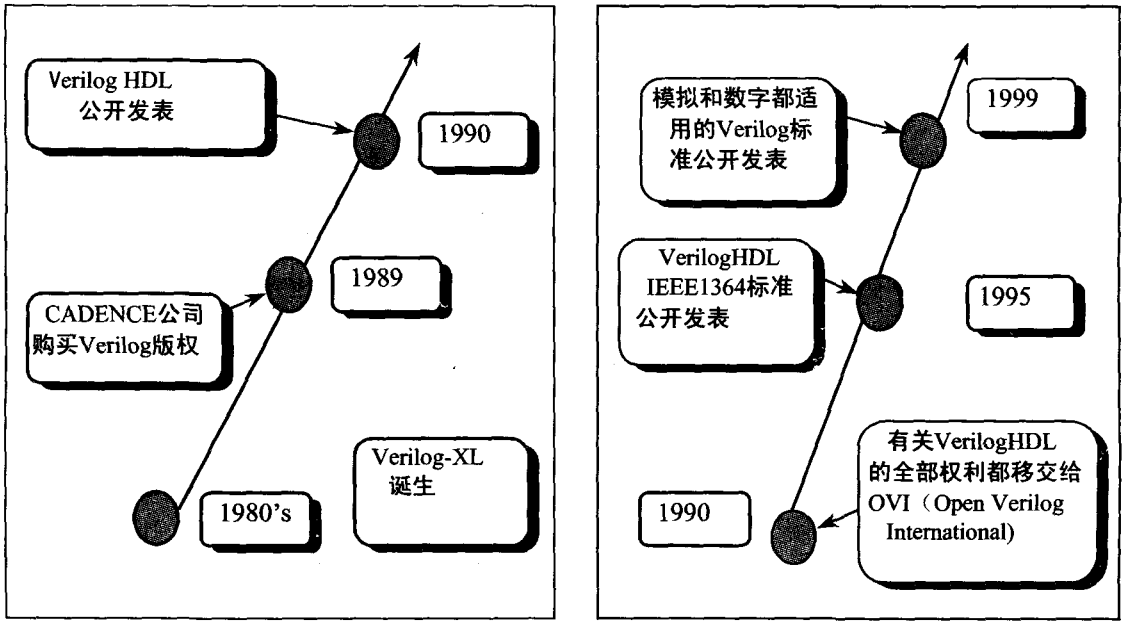

图 2.1　Verilog HDL 的历史和未来发展