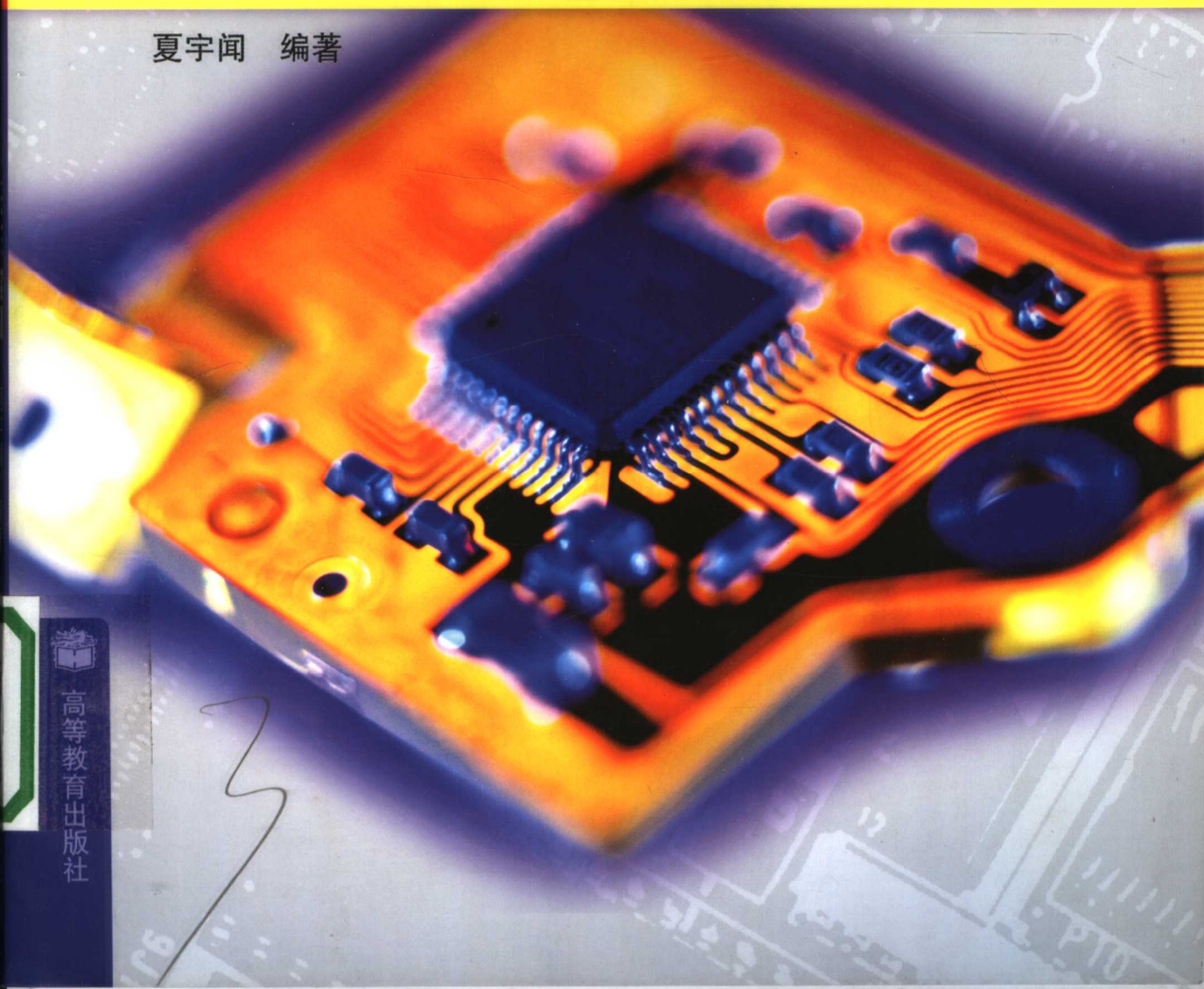
高 ◆ 等 ◆ 学 ◆ 校 ◆ 教 ◆ 材

数字系统设计

——Verilog 实现

夏宇闻 编著



高等教育出版社

高等学校教材

数字系统设计
——Verilog 实现

夏宇闻　编著

高等教育出版社

内容提要

本书是在《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)的基础上修订而成的。本书讲述利用 Verilog 硬件描述语言进行建模、仿真和综合设计复杂数字系统的方法。全书在介绍现代数字系统设计方法思想的基础上，深入地讲解了常用的 Verilog HDL（IEEE 1364—2001 标准）语法，通过一系列由浅入深的设计示例，剖析了数字系统设计方法的核心，然后逐步过渡到工程设计范例的讲解。本书同时配有实验练习和语法手册，可供读者上机练习和查阅，以提高利用 Verilog HDL 设计复杂数字系统的能力。

本书可以作为高等学校电子信息、自动控制和计算机工程类的本科高年级和研究生的教学和实验用书，亦可供工程技术人员自学与参考。

图书在版编目(CIP)数据

数字系统设计——Verilog 实现/夏宇闻编著. —2 版.
北京：高等教育出版社，2006.1
ISBN 7-04-017198-8

Ⅰ. 数… Ⅱ. 夏… Ⅲ. ①数字系统－系统设计－高等学校－教材②硬件描述语言，VHDL－程序设计－高等学校－教材 Ⅳ. ①TP271②TP312

中国版本图书馆 CIP 数据核字（2005）第 148523 号

策划编辑 董建波　**责任编辑** 董建波　**封面设计** 张申申　**责任绘图** 朱　静
版式设计 马静如　**责任校对** 胡晓琪　**责任印制** 杨　明

出版发行	高等教育出版社	购书热线	010 - 58581118
社　址	北京市西城区德外大街 4 号	免费咨询	800 - 810 - 0598
邮政编码	100011	网　址	http://www.hep.edu.cn
总　机	010 - 58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
			http://www.landraco.com.cn
经　销	蓝色畅想图书发行有限公司	畅想教育	http://www.widedu.com
印　刷	北京嘉实印刷有限公司		
		版　次	2002年12月第1版
开　本	787×1092　1/16		2006年 1 月第 2 版
印　张	24.25	印　次	2006年 1 月第 1 次印刷
字　数	540 000	定　价	27.80 元

物料号　17198-00

前　言

《从算法设计到硬件逻辑的实现——复杂数字逻辑系统的 Verilog HDL 设计技术和方法》（夏宇闻编著）及其配套的实验指导书，自 2001 年由高等教育出版社出版以来，受到广大师生的欢迎。4 年多来，数字系统的 Verilog 设计方法逐渐受到教育界的重视，读者群体日益扩大。为了适应新技术的发展，本教材在原教材的基础上做了许多修改，补充了一些常用的 Verilog 2001 标准的新内容，添加了一些由浅入深的常用设计示例，再逐步过渡到工程设计范例的讲解。本书同时配有实验练习和语法手册，补充了更多由浅入深的练习和思考题。为了更好地反映教材的内容，将书名更改为《数字系统设计——Verilog 实现》。这套教材是笔者十多年来在数字系统 Verilog 设计教学方面的经验总结。

为了使不同层次的读者都有所收获，本书的前八章可以作为本科生的入门教材，第九章到第十八章可以作为本科高年级学生或研究生学习数字系统设计的参考，也可以作为有数字电路基础知识的相关从业人员的自学教材。每章后都有总结和思考题帮助读者复习所学内容。本书的配套参考书是为想真正掌握 Verilog 设计方法的读者精心设计的上机练习和范例，并附有常用的语法供参考，能有效地帮助读者理解课堂讲解的知识。

编　者

2005 年 8 月

目　录

第一篇　绪　论

第二篇　基 础 部 分

第三篇 设计和验证部分

第一篇 绪 论

我们都学习过数字电路基础，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、电阻等器件构成的，能执行相应的开关逻辑操作。由与门、或门和非门又可以构成各种触发器，它们可以记忆状态。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运行原理后，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。这些基础知识使我们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。但如何真正地设计一个复杂的数字系统，以及如何验证我们设计的逻辑系统功能是否正确，还缺少好的手段和方法。本书主要讲解用 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。下面先就复杂数字系统的概念、用途和几个有关的基本问题做一下说明。

1．设计专用复杂数字系统的原因

现代计算机与通信系统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。这就是人们常用 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原因。

在数字信号处理领域内有相当大的一部分工作是可以事后处理的。人们可以利用通用的计算机系统来处理这类问题。如在石油地质调查中人们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，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，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。很难想象能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。人们必须为这样的运算设计专用的高速硬线逻辑电路，这可以在高速 FPGA 器件上实现或制成高速专用集成电路来实现。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的控制下，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的目的而设计的，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。

本书是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，使读者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，并向读者展示一种 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，对数据的输入/输出接口有明确的理解，完全有能力设计并制造出有自己知识产权的数字信号处理（DSP）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．实时数字信号处理的实现方法

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均得到了迅速的发展。这两个表面上看起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。而数据的输入/输出与通用的计算机系统不同，往往不是文件系统，而是高速的数据码流。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，才有可能实现真正意义上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工具的基础。

3．有关数字信号处理的几个基本概念

1）计算

说到数字信号处理，人们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。其基本思路是先把信号用一系列的数字表示，如是连续的模拟信号，则需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的码流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（也可以称为滤波），有的是为了压缩数据以减少占用的频道……有时人们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。它提出的最基本的问题是什么样的工作能自动完成，什么样的不能。”①

本书中提到的“计算”这个词，指的就是 Computing 所包含的意思。由传统的观点出发，可以从数学、科学和工程这 3 个不同的方面来研究计算。由比较现代的观点出发，可以从算法和数据结构、编程语言、体系结构、软件和硬件设计方法学这 4 个主要的方面来研究计算，

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，因此本书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，并特别强调利用 Verilog 硬件描述语言的 Top-Down 设计方法的介绍。

2）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，人们需要采用一些特殊的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模型。

3）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。人们已经熟悉通过编写程序来解决计算问题，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，无论是电路系统的仿真还是综合都需要掌握硬件描述语言。本书将比较详细地介绍 Verilog 硬件描述语言。

4）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）在 1946 年设计的 EDVAC 电子计算机，它的结构是一种最早的顺序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

① Denning et al., “Computing as a Discipline” Communication of ACM, January,1989

论和研究如何提高 CPU 运算速度和性能的学问。对 CPU 微体系结构的了解是设计高性能的专用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 进行复杂数字电路设计的技术和方法，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，可以参阅其他相关书籍。

4．设计方法学

复杂数字系统的设计是一个把思想（即算法和通信接口的协议）转化为实际数字逻辑电路的过程。众所周知，同一个算法和协议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可以有很大的差别。人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，下面是常用的 4 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（从几万门到几百万门）；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。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来确定采用哪种方案。

在上述第②、③、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。

5．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。人们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，把该 C 语言程序转换为专用微处理器的汇编再编译为机器代码，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，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数字信号处理（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某一种特殊应用，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较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 在需要高速部分进行增强，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、投资成本高，往往只有经费充足、批量大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。

近年来，基于平台（Platform-Based Design，PBD）的设计方法学已经开始在先进的电子工业国家普及。所谓 PBD 方法，实际上就是把许多经过充分验证的现成的 IP 核集中到一个 Verilog 验证平台上，供设计者选用，所以能大大提高系统芯片（System on Chip，SoC）的设计效率。这些现成的 IP 核不但包括各种档次的通过配置可以裁剪功能的处理器内核，还包括各种有关通信协议、图像处理和接口硬件的可以根据需求裁剪的 IP 核，以及一些配套的、可以在处理器上运行的现成程序。这些硬/软 IP 核的 Verilog 行为模型都可以通过 Verilog 模块在验证平台上体现。而制造所需要的技术资料，如逻辑结构和网表，也可以用门级 Verilog 模块表示。这些 IP 核的最后实现当然是由基本元器件的版图拼接而成的，它们的性能能够得到保证，并与其 IP 核的 Verilog 行为模型完全一致。无疑这种商业化的全社会协同运作大大加快了复杂 SoC 的设计效率。我国的有关部门和 SoC 工程技术人员必须理解 Verilog 设计方法的重要性，加强全国各设计单位的协作，否则我国电子设计行业赶上世界的发展步伐也只能是一句空话。

6．C 语言、Matlab 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、各种 EDA 软件工具。就编程语言而言，国内外大多数学校都以 C 语言为标准。而 Matlab 则是一个常用的数学计算软件包，其中有许多现成的数学函数供利用，大大节省了编程时间，Matlab 还提供与 C 程序模块进行方便地接口的手段，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，它能很快生成有用的数据文件和表格，并可以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，人们必须先深入了解 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，将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 语言程序外，还需编写硬件描述语言（如 Verilog HDL 或 VHDL）的程序进行仿

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。

使用硬件描述语言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 语言配合硬件描述语言来做逻辑设计布线前和布线后的仿真，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要考虑的 3 个主要问题：

（1）为什么选择 C 语言与 Verilog HDL 配合使用？

（2）C 语言与 Verilog HDL 的使用有何限制？

（3）如何利用 C 语言来加速硬件的设计和故障检测？

1）选择 C 语言与 Verilog 配合使用的原因

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 语言程序的设计环境比 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 语言程序一段一段地改写成用并型结构（类似于 Verilog）描述的 C 语言程序。此时还是在 C 语言的环境里，使用的依然是 C 语言。如果运行结果都正确，就将 C 语言的关键字用 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 语言与 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 HDL 在使用中的限制

在设计流程中，C 语言与 Verilog HDL 的混合使用往往在两种语言的转换中会遇到许多难题。例如，怎样把 C 语言程序转换成类似 Verilog 结构的 C 语言程序，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 语言中迭代、指针、不确定次数的循环等较抽象的语法也能来表示算法（因为转换的目的是要用可综合的 Verilog 语句来代替 C 语言程序中的语句，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 语言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。而且 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 语言的语法变化则很多，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 语言程序调用函数是没有延时的，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，也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）利用 C 语言来加快硬件的设计和查错的方法

表 0.1.1 中列出了常用的 C 语言与 Verilog 相对应的关键字与控制结构。

表 0.1.1 C 语言与 Verilog 相对应的关键字与控制结构

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

表 0.1.2 中列出了 C 语言与 Verilog 相对应的运算符及其功能。

上面的讨论可以总结如下：

（1）C 语言与 Verilog 硬件描述语言可以配合使用，以辅助设计硬件。

（2）C 语言与 Verilog 硬件描述语言很像，只要稍加限制，C 语言的程序就能很容易地转成 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，中国内地目前使用 Verilog HDL 与 VHDL 两者兼有。到底选用 VHDL 或是 Verilog HDL 来配合 C 语言一起用，可由读者自行决定。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。从使用的角度来看，支持 Verilog HDL 的半导体厂家也比支持 VHDL 的多。

表 0.1.2 C 语言与 Verilog 相对应的运算符号及其功能

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

总 结

在绪论中，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，向读者展示了一种 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主知识产权的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。在下面的各章里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 C 语言和硬件描述语言配合使用来提高设计效率？
7. 为什么 Verilog 语言能在系统芯片（SoC）的基于平台的设计方法中起核心作用？阐述 Verilog 语言对于 SoC 设计的重要性。