

第一部分 Verilog 基础知识

第 1 章 Verilog HDL 数字设计综述

第 2 章 层次建模的概念

第 3 章 基本概念

第 4 章 模块和端口

第 5 章 门级建模

第 6 章 数据流建模

第 7 章 行为级建模

第 8 章 任务和函数

第 9 章 实用建模技术

第 1 章 Verilog HDL 数字设计综述

1.1 数字电路 CAD 技术的发展历史

在过去的 20 多年中，数字电路设计技术的发展非常迅速。设计人员最早使用真空管和晶体管来设计数字电路。后来他们把逻辑门安置在单个芯片上，于是发明了集成电路。第一代集成电路（Integrated Circuit, IC）的门数非常少，称为小规模集成电路（Small Scale Integrated, SSI）。随着制造工艺技术的发展，设计者可以在单个芯片上布置数百个逻辑门，我们称之为中规模集成电路（Medium Scale Integrated, MSI）。随着大规模集成电路（Large Scale Integrated, LSI）的出现，数千个逻辑门能够集成在一起。设计过程由此开始变得非常复杂，因此设计者希望某些设计阶段能够自动完成。正是这种需要促进了电子设计自动化（Electronic Design Automation, EDA）的出现和发展^①。设计者开始使用电路和逻辑仿真技术对使用的基本组件的功能进行验证，这些基本组件的规模一般相当于几百个晶体管。不过这时的测试仍然在面包板上完成，设计人员在设计图纸或计算机图形终端上用手工完成电路的版图设计。

超大规模集成电路（Very Large Scale Integrated, VLSI）的出现，使得设计人员可以将超过 10 万个晶体管集成在一块芯片上。在这种情况下，已经不可能在面包板上对设计的功能进行验证了。计算机辅助技术对于超大规模集成电路的设计和验证变得非常重要，同时，使用计算机进行电路版图的布局和布线也开始流行，设计者在图形终端上用手工完成数字电路的门级设计。从小的功能模块开始设计，逐步使用小的功能模块来搭建高层功能模块，直到完成顶层设计。在最后制成芯片之前，设计者还会使用逻辑仿真工具对设计的功能进行验证。

随着设计规模的不断增大，其功能越来越复杂，逻辑仿真在整个设计过程中的作用越来越重要，使得设计者可以尽早地排除设计结构中存在的问题。

1.2 硬件描述语言的出现

很久以来，人们使用诸如 FORTRAN, Pascal, C 等语言来进行计算机程序设计，这些程序本质上是顺序执行的。同样，在硬件设计领域，设计人员也希望使用一种标准的语言来进行硬件设计。在这种情况下，许多种硬件描述语言（Hardware Description Languages, HDL）应运而生。设计者可以使用它对硬件中的并发执行过程建模。在出现的各种硬件描述语言中，Verilog HDL 和 VHDL 使用得最为广泛。Verilog HDL 于 1983 年源自 Gateway Design Automation 公司。稍后，由美国国防部的高级研究计划署牵头（制定合同）开发了 VHDL。设计人员很快认可了 Verilog HDL 和 VHDL 这两种语言，使用它们对大型数字电路进行仿真。

^① 本书第一版中使用了 CAD 工具这个术语。从技术角度看，CAD（Computer-Aided-Design，计算机辅助设计）工具这个术语指的是设计后端使用的工具，这些工具可以完成布局、布线和芯片的版图绘制等工作。而 CAE（Computer-Aided-Engineering，计算机辅助工程）工具这个术语指的是设计前端使用的工具，如 HDL 仿真、逻辑综合和时序分析。过去设计人员常把 CAD 和 CAE 这两个术语混用。目前，EDA 的范围包括了 CAD 和 CAE 两个部分。为了简单起见，在本书中我们把所有的设计工具都称为 EDA 工具。

虽然当时用 HDL 进行逻辑验证已经很普及,但是设计人员仍然需要手工将基于 HDL 的设计转换为由相互连接的逻辑门表示的电路简图。在 20 世纪 80 年代后期,逻辑综合工具的发展对数字电路的设计方法学产生了巨大的影响。设计者可以使用 HDL 在寄存器传输级(Register Transfer Level, RTL)对电路进行描述。在这种设计方法中,设计者只需要说明数据(信息)是如何在寄存器之间移动以及如何被处理的,而构成电路的逻辑门及其相互之间的连接数据(资料)由逻辑综合工具自动地从 RTL 描述中提取出来。

逻辑综合工具的出现和发展使得 HDL 在数字电路设计中占据了重要的地位。设计者不再需要通过手工用逻辑门来搭建电路。他们可以使用硬件描述语言来描述电路的功能和数据的流向,然后由逻辑综合工具自动综合出由逻辑门及其相互连接构成的电路结构细节,实现 HDL 所描述和指定的特定功能。

同样, HDL 在系统级设计中也得到了应用。HDL 用来仿真电路板、互连总线、FPGA (Field Programmable Gate Arrays) 以及 PAL (Programmable Array Logic) 等。通常的方法是使用 HDL 单独设计每个芯片,然后通过仿真来验证整个系统的功能。

目前, Verilog HDL 已经是公认的 IEEE 标准。Verilog HDL 的第一个标准 (IEEE 1364-1995) 是在 1995 年批准的。最近公布的 IEEE 1364-2001 标准与原标准相比有了显著的改进。

1.3 典型设计流程

图 1.1 表示的是超大规模集成电路设计的典型流程。图中不带阴影的方框表示设计描述的层次,带阴影的方框表示设计的过程。

在任何设计流程中,必须首先编写设计电路的技术指标和功能要求细节,从抽象的角度对电路的功能、接口和总体结构进行描述,在这一阶段无须考虑电路的具体实现方式。接下来设计者使用行为级描述来分析电路的功能、性能、标准兼容性以及其他高层次的问题。行为级描述一般也使用 HDL 来编写^①。

由行为级描述向 RTL 级描述的转换是由设计者手工完成的。在这个过程中,设计者需要对实现电路功能的数据流进行详细描述。在以后的各个设计步骤中,设计者都可以借助工具软件。

综合工具的作用是将 RTL 级描述转换成门级网表。门级网表从逻辑门及其相互连接关系的角度来描述电路的结构。综合工具需要保证综合出来的门级网表满足时序、面积以及功耗的要求。自动布局、布线工具读入综合得到的网表并生成电路的版图。电路的版图经过验证就可以制成芯片。

在整个设计流程中,设计的重点主要在于手工对 RTL 描述的优化。在 RTL 描述完成之后,设计者就可以在 EDA 工具的辅助下完成后续的设计过程。从 RTL 级描述着手可以将设计周期从几年缩短为几个月,也有可能较短的时间内对设计进行多次调整和改进。

最近出现的行为级综合工具可以将电路或算法的行为级描述转换为 RTL 级描述。随着行为级综合工具的不断完善,数字电路设计将越来越类似于软件程序设计。设计者只需在非常抽象的层次使用 HDL 对电路的算法进行描述,然后在 EDA 工具的帮助下完成从行为级描述到最终芯片结构的转换。

^① 新出现的 EDA 工具可以对电路的行为描述进行仿真。这些新工具将 HDL 和面向目标的编程语言(如 C++等)的强大功能结合在一起。有了这些工具,则不需要使用 Verilog HDL 来编写电路的行为模型。

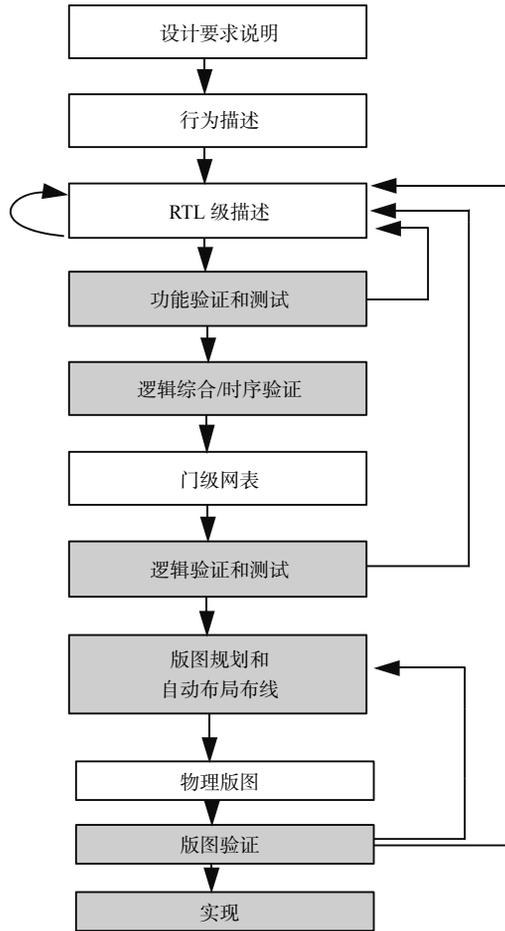


图 1.1 典型的设计流程

虽然 EDA 工具能够使设计过程自动化并显著缩短设计周期，但设计者仍然是整个设计过程的核心。如果使用不当，EDA 工具也会产生很差的结果。因此设计者还必须对这种设计方法的缺陷有所了解，才能借助于 EDA 工具得出优化的设计。

1.4 硬件描述语言的意义

与传统的基于电路原理图的设计方法相比，使用硬件描述语言 (HDL) 进行设计具有许多优点：

- 通过使用 HDL，设计者可以在非常抽象的层次上对电路进行描述。设计者可以在 RTL 级对电路进行描述而不必选择特定的制造工艺，逻辑综合工具能够将设计自动转换为任何一种制造工艺版图。如果出现新的制造工艺，设计者不必对电路进行重新设计，只需将 RTL 级描述输入逻辑综合工具，即可生成针对新工艺的门级网表。逻辑综合工具将根据新的工艺对电路的时序和面积进行优化。
- 通过使用 HDL，设计者可以在设计周期的早期对电路的功能进行验证。设计者可以很容易地对 RTL 描述进行优化和修改，满足电路功能的要求。由于能够在设计初期发现和排除绝大多数设计错误，因此大大降低了在设计后期的门级网表或物理版图上出现错误的可能性，避免了设计过程的反复，显著地缩短了设计周期。

- 使用 HDL 进行设计类似于编写计算机程序，带有文字注释的源程序非常便于开发和修改。与门级电路原理图相比，这种设计表达方式能够对电路进行更加简明扼要的描述。非常复杂的设计，如果用门级电路原理图来表达，几乎是无法理解的。

基于 HDL 语言的设计方法是不可动摇的^①。随着数字电路复杂性的不断增加以及 EDA 工具的日益成熟，基于硬件描述语言的设计方法已经成为大型数字电路设计的主流。没有一个数字电路设计师能承担无视这种新设计方法所付出的代价。

1.5 Verilog HDL 的优点

Verilog HDL 已经发展成为标准的硬件描述语言。对于硬件设计，它具有许多优点：

- Verilog HDL 是一种通用的硬件描述语言，易学易用。由于它的语法与 C 语言类似，因此对于具有 C 语言编程经验的设计者来说，很容易学习和掌握。
- Verilog HDL 允许在同一个电路模型内进行不同抽象层次的描述。设计者可以从开关、门、RTL 或者行为等各个层次对电路模型进行定义。同时，设计者只需要学习一种语言就能够使用它来描述电路的激励，进行层次化设计。
- 绝大多数流行的综合工具都支持 Verilog HDL，这是 Verilog HDL 成为设计者的首选语言的重要原因之一。
- 所有的制造厂商都提供用于 Verilog HDL 综合之后的逻辑仿真的元件库，因此使用 Verilog HDL 进行设计，即可在更广泛的范围内选择委托制造的厂商。
- 编程语言接口（PLI）是 Verilog 语言最重要的特性之一，它使得设计者可以通过自己编写 C 代码来访问 Verilog 内部的数据结构。设计者可以使用 PLI 按照自己的需要来配置 Verilog HDL 仿真器。

1.6 硬件描述语言的发展趋势

数字电路速度和复杂性正在迅速地增长，这就要求设计者从更高的抽象层次对电路进行描述。这样的好处是设计者只需从功能的角度进行设计，由 EDA 工具来完成具体实现细节。在设计者的指导下，EDA 工具可以完成非常复杂的从设计到实现的转换，并且达到近似的优化效果。

由于逻辑综合工具可以从 RTL 描述生成门级网表，目前基于 HDL 的主流设计方式是 RTL 级设计。行为级综合工具允许直接对电路的算法和行为进行描述，然后由 EDA 工具在各个设计阶段进行转换和优化，不过这种设计方式尚未被业界广泛接受。同时，Verilog HDL 本身也在不断地补充和完善，以适应新的设计验证方法。

形式验证和断言检查（formal verification and assertion checking）是最近发展起来的设计验证方法。形式验证使用形式化的数学方法来验证 Verilog HDL 描述的正确性，并且对 RTL 描述与综合后得到的门级网表电路行为的等价性进行检查。然而，用 Verilog HDL 对设计进行描述的需求将不会消失。断言检查允许将检查规嵌入到 RTL 描述中，以便于对设计中最重要部分进行检查。

^① 在过去的几年中已经出现了几种主要用于验证的新工具和新语言。这些新语言比较适合进行功能验证。但是，就逻辑设计而言，HDL 还是优先选择的语言，目前是不可替代的。

与此相适应,新的面向验证的语言也获得了快速的发展。这些语言既包含 HDL 中的并发特性和其他用于描述硬件的语法结构,又具有 C++面向对象的特点,同时还具有测试激励自动生成、性能检查和代码覆盖的特性。但是,这些语言并不是 Verilog HDL 的替代者,其目的仅在于大大缩短验证时间。设计描述仍然需要使用 Verilog HDL。

对于像微处理器这样的超高速、对时序有着严格要求的电路来说,逻辑综合工具生成的门级网表并不是优化的。在这种情况下,设计者经常需要直接在 RTL 描述中嵌入门级描述,以达到优化的效果。由于设计者希望尽最大可能提高电路的速度,而 EDA 工具有时并不能满足这种要求,因此虽然这种嵌入门级描述的方法并不符合高层次的设计流程,但是却被设计者经常采用。

系统级设计采用的另一种技术是结合使用自底向上的方法。设计者通过使用现有的 Verilog HDL 模块、基本功能块或者第三方提供的核心功能块来快速搭建系统,以便进行仿真。这种方法降低了开发费用,缩短了开发周期。例如,考虑设计一个含有 CPU、图形处理芯片、I/O 芯片和系统总线的系统。CPU 的设计者自己着手从 RTL 级开始开发下一代 CPU。为了能够在设计前期对 CPU 进行系统级仿真,设计者可以使用图形处理芯片和 I/O 芯片的行为级模型和第三方提供的系统总线模型来构建系统,这样就可以在图形处理芯片和 I/O 芯片的 RTL 描述完成之前迅速地进行仿真,达到降低开发费用、缩短开发周期的目的。