第1章 绪 论

1.1 EDA技术的含义

由于 EDA 是一门迅速发展的新技术，涉及面广，内容丰富，理解各异，因此目前尚无统一的看法。比较一致的看法是：EDA 技术，是以大规模可编程逻辑器件为设计载体，以硬件描述语言为系统逻辑描述的主要表达方式，以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具，通过有关的开发软件，自动完成用软件的方式设计的电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真，直至完成对于特定目标芯片的适配编译、逻辑映射、编程下载等工作，最终形成集成电子系统或专用集成芯片的一门新技术。

利用 EDA 技术设计电子系统具有以下几个特点：① 用软件的方式设计硬件；② 用软件方式设计的系统到硬件系统的转换是由有关的开发软件自动完成的；③ 设计过程中可用有关软件进行各种仿真；④ 系统可现场编程，在线升级；⑤ 整个系统可集成在一个芯片上，体积小、功耗低、可靠性高。因此，EDA 技术是现代电子设计的发展趋势。

1.2 EDA技术的发展历程

伴随着计算机、集成电路、电子系统设计的发展，EDA 技术经历了计算机辅助设计（Computer Assist Design，CAD）、计算机辅助工程设计（Computer Assist Engineering Design，CAE）和电子设计自动化（Electronic Design Automation，EDA）三个发展阶段。

1．20 世纪 70 年代的计算机辅助设计阶段

早期的电子系统硬件设计采用的是分立元件，随着集成电路的出现和应用，硬件设计进入发展的初级阶段。初级阶段的硬件设计大量选用中小规模标准集成电路，人们将这些器件焊接在电路板上，做成初级电子系统，对电子系统的调试是在组装好的印制电路板（Printed Circuit Board，PCB）上进行的。

由于设计师对图形符号使用数量有限，传统的手工布图方法无法满足产品复杂性的要求，更不能满足工作效率的要求。这时，人们开始将产品设计过程中高度重复性的繁杂劳动，如布图、布线工作，用二维图形编辑与分析的 CAD 工具替代，其中最具代表性的 CAD 产品是美国 ACCEL 公司开发的 Tango 布线软件。20 世纪 70 年代是 EDA 技术发展初期，由于 PCB 布图、布线工具受到计算机工作平台的制约，其支持的设计工作有限，且性能较差。

2．20 世纪 80 年代的计算机辅助工程设计阶段

初级阶段的硬件设计是用大量不同型号的标准芯片实现电子系统设计的。随着微电子工艺的发展，相继出现了集成上万个晶体管的微处理器、集成几十万直到上百万个存储单元的随机存储器和只读存储器。此外，支持定制单元电路设计的硅编辑、掩膜编程的门阵列，如标准单

元的半定制设计方法及可编程逻辑器件（PAL 和 GAL）等一系列微结构和微电子学的研究成果，都为电子系统的设计提供了新天地。因此，可以用少数几种通用的标准芯片实现电子系统的设计。

伴随计算机和集成电路的发展，EDA 技术进入计算机辅助工程设计阶段。20 世纪 80 年代初，推出的 EDA 工具则以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心，重点解决电路设计没有完成之前的功能检测等问题。利用这些工具，设计师能在产品制作之前预知产品的功能与性能，能生成产品制造文件，因此在设计阶段对产品性能的分析前进了一大步。

如果说 20 世纪 70 年代的自动布局、布线的 CAD 工具代替了设计工作中绘图的重复劳动，那么 20 世纪 80 年代出现的具有自动综合能力的 CAE 工具则代替了设计师的部分工作，对保证电子系统的设计，制造出最佳的电子产品起着关键的作用。20 世纪 80 年代后期，EDA 工具已经可以进行设计描述、综合与优化和设计结果验证，CAE 阶段的 EDA 工具不仅为成功开发电子产品创造了有利条件，而且为高级设计人员的创造性劳动提供了方便。但是，大部分从原理图出发的 EDA 工具仍然不能适应复杂电子系统的设计要求，而具体化的元件图形制约着优化设计。

3．20 世纪 90 年代电子系统设计自动化阶段

为了满足千差万别的系统用户提出的设计要求，最好的办法是由用户自己设计芯片，让他们把想设计的电路直接设计在自己的专用芯片上。微电子技术的发展，特别是可编程逻辑器件的发展，使得微电子厂家可以为用户提供各种规模的可编程逻辑器件，使设计者通过设计芯片实现电子系统功能。EDA 工具的发展，又为设计师提供了全线 EDA 工具。这个阶段发展起来的 EDA 工具，目的是在设计前期将设计师从事的许多高层次设计由工具来完成，例如可以将用户要求转换为设计技术规范，有效地处理可用设计资源与理想设计目标之间的矛盾，按具体的硬件、软件和算法分解设计等。由于电子技术和 EDA 工具的发展，设计师可以在不太长的时间内使用 EDA 工具，通过一些简单标准化的设计过程，利用微电子厂家提供的设计库来完成数万门 ASIC 和集成系统的设计与验证。

20 世纪 90 年代，设计师逐步从使用硬件转向设计硬件，从单个电子产品开发转向系统级电子产品开发（即片上系统集成，System On a Chip）。因此，EDA 工具是以系统及设计为核心，包括系统行为级描述与结构综合、系统仿真与测试验证、系统划分与指标分配、系统决策与文件生成等一整套电子系统设计自动化工具。这时的 EDA 工具不仅具有电子系统设计的能力，而且能提供独立于工艺和厂家的系统级设计能力，具有高级抽象的设计构思手段。例如，具有提供方框图、状态图和流程图的编辑能力，具有适合层次描述和混合信号描述的硬件描述语言（如 VHDL、AHDL 或 Verilog-HDL），同时含有各种工艺的标准元件库。

只有具备上述功能的 EDA 工具，才可能使电子系统工程师在不熟悉各种半导体工艺的情况下，完成电子系统的设计。

未来的 EDA 技术将向广度和深度两个方向发展，EDA 将会超越电子设计的范畴进入其他领域，随着基于 EDA 的 SoC（片上系统）设计技术的发展、软硬核功能库的建立，以及基于 VHDL 的自顶向下设计理念的确立，未来的电子系统的设计与规划将不再是电子工程师的专利。

1.3　EDA 技术的主要内容

EDA 技术涉及面广，内容丰富。从教学和实用的角度看，应主要掌握 4 方面的内容：

① 大规模可编程逻辑器件。

② 硬件描述语言。

③ 软件开发工具。

④ 实验开发系统。

其中，大规模可编程逻辑器件是利用 EDA 技术进行电子系统设计的载体，硬件描述语言是利用 EDA 技术进行电子系统设计的主要表达手段，软件开发工具是利用 EDA 技术进行电子系统设计的智能化的自动化设计工具，实验开发系统则是利用 EDA 技术进行电子系统设计的下载工具及硬件验证工具。

为了使读者对 EDA 技术有一个总体印象，下面简要介绍 EDA 技术的主要内容。

1．大规模可编程逻辑器件

可编程逻辑器件（Programmable Logic Device，PLD）是一种由用户编程以实现某种逻辑功能的新型逻辑器件。FPGA 和 CPLD 分别是现场可编程门阵列和复杂可编程逻辑器件的简称。今天，FPGA 和 CPLD 器件的应用已十分广泛，它们将随着 EDA 技术的发展而成为电子设计领域的重要角色。国际上生产 FPGA/CPLD 且在我国市场份额较大的主流公司是 Xilinx、Altera、Lattice。Xilinx 公司的 FPGA 器件有 XC2000、XC3000、XC4000、XC4000E、XC4000XLA、XC5200 系列等，可用门数为 1200～18000；Altera 公司的 CPLD 器件有 FLEX6000、FLEX8000、FLEX10K、FLEX10KE 系列等，可用门数为 5000～25000；Lattice 公司的 ISP-PLD 器件有 ispLSI1000、ispLSI2000、ispLSI3000、ispLSI6000 系列等，集成度可多达 25000 个 PLD 等效门。

FPGA 在结构上主要分为三部分，即可编程逻辑单元、可编程输入/输出单元和可编程连线。CPLD 在结构上主要包括三部分，即可编程逻辑宏单元、可编程输入/输出单元和可编程内部连线。

高集成度、高速度和高可靠性是 FPGA/CPLD 最明显的特点，其时钟延时可小至纳秒级，结合其并行工作方式，在超高速应用领域和实时测控方面有着非常广阔的应用前景。在高可靠应用领域，如果设计得当，将不会存在类似于 MCU 的复位不可靠和 PC 可能跑飞等问题。FPGA/CPLD 的高可靠性还表现在几乎可将整个系统下载到同一芯片中，实现所谓的片上系统，从而大大缩小体积，易于管理和屏蔽。

由于 FPGA/CPLD 的集成规模非常大，因此可利用先进的 EDA 工具进行电子系统设计和产品开发。由于开发工具的通用性、设计语言的标准化以及设计过程几乎与所用器件的硬件结构无关，因而设计开发成功的各类逻辑功能块软件有很好的兼容性和可移植性。它几乎可用于任何型号和规模的 FPGA/CPLD 中，因此可使得产品设计效率大幅提升。可以在很短时间内完成十分复杂的系统设计，这正是产品快速进入市场最宝贵的特征。美国的 IT 公司认为，ASIC 百分之八十的功能可用于 IP 核等现有逻辑合成，而未来大系统的 FPGA/CPLD 设计仅是各类再应用逻辑与 IP 核的拼装，其设计周期将更短。

与 ASIC 设计相比，FPGA/CPLD 的明显优势是开发周期短、投资风险小、产品上市速度快、市场适应能力强和硬件升级回旋余地大，而且当产品定型和产量扩大后，可将在生产中达到充分检验的 VHDL 设计迅速实现 ASIC 投产。

对于一个开发项目，究竟是选择 FPGA 还是选择 CPLD 呢？主要看开发项目本身的需要。对于普通规模且产量不是很大的产品项目，通常使用 CPLD 较好。对于大规模的逻辑设计或单片系统设计，则多采用 FPGA。另外，FPGA 掉电后将丢失原有的逻辑信息，所以在实用中需要为 FPGA 芯片配置一个专用 ROM。

2．硬件描述语言（HDL）

常用的硬件描述语言有 VHDL、Verilog 和 ABEL。

VHDL：作为 IEEE 的工业标准硬件描述语言，VHDL 在电子工程领域已成为事实上的通用硬件描述语言。

Verilog：支持的 EDA 工具较多，适用于 RTL 级和门电路级的描述，其综合过程较 VHDL 稍简单，但其在高级描述方面不如 VHDL。

ABEL：这是一种支持各种不同输入方式的 HDL，广泛用于各种可编程逻辑器件的逻辑功能设计。由于其语言描述的独立性，因而适用于各种不同规模的可编程器件的设计。

3．软件开发工具

目前比较流行的 EDA 的软件工具有 Altera 公司的 MAX+plus II 和 Quartus、Lattice 公司的 IspEXPERT，以及 Xilinx 公司的 ISE。

MAX+plus II：支持原理图、VHDL 和 Verilog 语言文本文件，它以波形与 EDIF 等格式的文件作为设计输入，支持这些文件的任意混合设计。它具有门级仿真器，可以进行功能仿真和时序仿真，能够产生精确的仿真结果。在适配之后，MAX+plus II 可生成供时序仿真用的 EDIF、VHDL 和 Verilog 三种不同格式的网表文件，界面友好，使用简单，是业界最易学、易用的 EDA 软件。它还支持主流的第三方 EDA 工具，支持除 APEX20K 系列之外所有 Altera 公司的 FPGA/CPLD 大规模逻辑器件。

IspEXPERT：IspEXPERT System 是 IspEXPERT 的主要集成环境。通过它可以进行 VHDL、Verilog 及 ABEL 语言的设计输入、综合、适配、仿真和系统下载。IspEXPERT System 是目前流行 EDA 软件中最易掌握的设计工具之一，其界面友好、操作方便、功能强大，与第三方 EDA 工具兼容良好。

ISE：Xilinx 公司最新集成开发的 EDA 工具。它采用自动化的、完整的集成设计环境。ISE 设计套件 10.1 是 Xilinx 推出的业内领先设计工具的最新版本，提供了完美的设计性能和生产率组合。

4．实验开发系统

提供芯片下载电路及 EDA 实验/开发的外围资源（类似用于单片机开发的仿真器），供硬件验证用。一般包括：① 实验或开发所需的各类基本信号发生模块，包括时钟、脉冲、高低电平等；② FPGA/CPLD 输出信息显示模块，包括数码显示、发光管显示、声响指示等；③ 监控程序模块，提供“电路重构软配置”；④ 目标芯片适配座及上面的 FPGA/CPLD 目标芯片和编程下载电路。

1.4　EDA 软件系统的构成

EDA 技术研究的对象是电子设计的全过程，包括系统级、电路级和物理级 3 个层次的设计。它涉及的电子系统从低频、高频到微波，从线性到非线性，从模拟到数字，从通用集成电路到专用集成电路构造的电子系统，因此 EDA 技术研究的范畴相当广泛。如果从专用集成电路（ASIC）开发与应用角度看，EDA 软件系统应当包含以下子模块：设计输入子模块、设计数据库子模块、分析验证子模块、综合仿真子模块、布局布线子模块等。

（1）设计输入子模块：该模块接受用户的设计描述，并进行语义正确性、语法规则的检查，检查通过后，将用户的设计描述数据转换为 EDA 软件系统的内部数据格式，存入设计数据库供其他子模块调用。设计输入子模块不仅能接受图形描述输入、硬件描述语言（HDL）描述输入，还能接受图文混合描述输入。该子模块一般包含针对不同描述方式的编辑器，如图形编辑器、文本编辑器等，同时包含对应的分析器。

（2）设计数据库子模块：该模块存放系统提供的库单元，以及用户的设计描述和中间设计结果。

（3）分析验证子模块：该模块包括各个层次的模拟验证、设计规则的检查、故障诊断等。

（4）综合仿真子模块：该模块包括各个层次的综合工具，理想的情况是：从高层次到低层次的综合仿真全部由 EDA 工具自动实现。

（5）布局布线子模块：该模块实现由逻辑设计到物理实现的映射，因此与物理实现的方式密切相关。例如，最终的物理实现可以是门阵列、可编程逻辑器件等。由于对应的器件不同，因此各自的布局布线工具会有很大的差异。

近年来，许多生产可编程逻辑器件的公司都相继推出了适于开发自己公司器件的 EDA 工具，这些工具一般都具有上面提到的各个模块，操作简单，对硬件环境要求低，运行平台是 PC 和 Windows 或 Windows NT 操作系统。例如，Xilinx、Altera、Lattice、Actel、AMD 等器件公司都有自己的 EDA 工具。

EDA 工具不只面向 ASIC 的应用与开发，还有涉及电子设计各个方面的 EDA 工具，包括数字电路设计、模拟电路设计、数模混合设计、系统设计、仿真验证等电子设计的许多领域。这些工具对硬件环境要求高，一般运行平台要求是工作站和 UNIX 操作系统，功能齐全、性能优良，一般由专门开发 EDA 软件工具的软件公司提供，如 Cadence、Mentel Graphics、Viewlogic、Synopsys 等软件公司都有其特色工具。

Viewlogic 公司的 EDA 工具就有基本工具、系统设计工具和 ASIC/FPGA 设计工具三大类，共 20 多个工具。

基本工具包括：原理图输入工具 ViewDraw，数字仿真器 ViewSim，波形编辑与显示器 ViewTrace，静态时序分析工具 Motive，设计流程管理工具 ViewFlow。

系统设计工具包括：模拟电路仿真器 ViewSpice，PLD 开发工具包 ViewPLD，库开发工具 ViewLibrarian，PCB 信号串扰分析工具 XTK，PCB 布线前信号分析工具 PDQ，电磁兼容设计工具 QUIET，PCB 版面规划工具 ViewFloorplanner。

ASIC/FPGA 设计工具包括：VHDL 仿真器 SpeedWave，SpeedWave Verilog 仿真器 VCS，

逻辑综合工具 ViewSynthesis，自动测试向量生成工具 Test Gen/Sunrise，原理图自动生成工具 ViewGen，有限状态机设计工具 ViewFSM，Datapath 设计工具 ViewDatapath，VHDL 与 Verilog 混合仿真环境 FusionHDL。

1.5 EDA 工具的发展趋势

1．设计输入工具的发展趋势

早期 EDA 工具设计输入普遍采用原理图输入方式，以文字和图形作为设计载体和文件，将设计信息加载到后续的 EDA 工具中，完成设计分析工作。原理图输入方式的优点是直观，能满足以设计分析为主的一般要求，但原理图输入方式不适合用 EDA 综合工具。20 世纪 80 年代末，电子设计开始采用新的综合工具，设计描述开始由原理图设计描述转向以各种硬件描述语言为主的编程方式。用硬件描述语言描述设计，更接近系统行为描述，且便于综合，更适于传递和修改设计信息，还可以建立独立于工艺的设计文件；不便之处是不太直观，要求设计师学会编程。

很多电子设计师都具有原理图设计的经验，而不具有编程经验，所以仍然希望继续在比较熟悉的符号与图形环境中完成设计，而不是利用编程完成设计。为此，EDA 公司在 20 世纪 90 年代相继推出了一批图形化免编程的设计输入工具，它们允许设计师用其最方便并熟悉的设计方式，如框图、状态图、真值表和逻辑方程建立设计文件，然后由 EDA 工具自动生成综合所需的硬件描述语言文件。

2．具有混合信号处理能力的 EDA 工具

目前，数字电路设计的 EDA 工具远比模拟电路的 EDA 工具多，模拟集成电路 EDA 工具开发的难度较大，但由于物理量本身多以模拟形式存在，所以实现高性能的复杂电子系统的设计离不开模拟信号。因此，20 世纪 90 年代以来，EDA 工具厂商都比较重视数模混合信号设计工具的开发。对数字信号的语言描述，IEEE 已经制定了 VHDL 标准，而对模拟信号的语言正在制定 AHDL 标准。此外，还提出了适用于微波信号的 MHDL 描述语言。

具有混合信号设计能力的 EDA 工具能处理含有数字信号处理、专用集成电路宏单元、数模转换和模数转换模块、各种压控振荡器在内的混合系统设计。美国 Cadence、Synopsys 等公司开发的 EDA 工具已经具有混合设计能力。

3．更为有效的仿真工具的发展

通常，可以将电子系统设计的仿真过程分为两个阶段：设计前期的系统级仿真和设计过程的电路级仿真。系统级仿真主要验证系统的功能；电路级仿真主要验证系统的性能，决定怎样实现设计所需的精度。在整个电子设计过程中，仿真是花费时间最多的工作，也是占用 EDA 工具资源最多的一个环节。通常，设计活动的大部分时间是进行仿真，如验证设计的有效性、测试设计的精度、处理和保证设计要求等。仿真过程中仿真收敛的快慢同样是关键因素之一。要提高仿真的有效性，一方面是建立合理的仿真算法，另一方面是系统级仿真中系统级模型的建模，电路级仿真中电路级模型的建模。预计在下一代 EDA 工具中，仿真工具将有较大的发展。

4．更为理想的设计综合工具的开发

今天，电子系统和电路的集成规模越来越大，几乎不可能直接面向版图做设计，若要找出版图中的错误，更是难上加难。将设计者的精力从烦琐的版图设计和分析中转移到设计前期的算法开发和功能验证上，是设计综合工具要达到的目标。高层次设计综合工具可以将低层次的硬件设计一起转换到物理级的设计，实现不同层次、不同形式的设计描述转换，通过各种综合算法实现设计目标所规定的优化设计。当然，设计者的经验在设计综合中仍将起重要的作用，自动综合工具将有效地提高优化设计效率。

设计综合工具由最初的只能实现逻辑综合，逐步发展到可以实现设计前端的综合，直到设计后端的版图综合，以及测试综合的理想且完整的综合工具。设计前端的综合工具，可以实现从算法级的行为描述到寄存器传输级结构描述的转换，给出满足约束条件的硬件结构。在确定寄存器传输结构描述后，由逻辑综合工具完成硬件的门级结构的描述，逻辑综合的结果将作为版图综合的输入数据，进行版图综合。版图综合则是将门级和电路级的结构描述转换成物理版图的描述，版图综合时将通过自动交互的设计环境，实现按面积、速度和功率完成布局、布线的优化，实现最佳的版图设计。人们希望将设计测试工作尽可能地提前到设计前期，以便缩短设计周期，减少测试费用，因此测试综合贯穿于设计过程的始终。测试综合时可以消除设计中的冗余逻辑，诊断不可测的逻辑结构，自动插入可测性结构，生成测试向量；整个电路设计完成时，测试设计也随之完成。

面对当今飞速发展的电子产品市场，电子设计人员需要更加实用、快捷的 EDA 工具，使用统一的集成化设计环境，改变传统设计思路，即优先考虑具体物理实现方式，将精力集中到设计构思、方案比较和寻找优化设计等方面，以最快的速度开发出性能优良、质量一流的电子产品。今天的 EDA 工具将向着功能强大、简单易学、使用方便的方向发展。

1.6　EDA 的工程设计流程

1．源程序的编辑和编译

利用 EDA 技术进行一项工程设计，首先需要利用 EDA 工具的文本编辑器或图形编辑器，将它用文本方式或图形方式表达出来，进行排错编译，变成 VHDL 文件格式，为进一步的逻辑综合做好准备。

常用的源程序输入方式有如下三种。

（1）原理图输入方式：利用 EDA 工具提供的图形编辑器以原理图的方式进行输入。原理图输入方式比较容易掌握，直观且方便，所画的电路原理图（注意，这种原理图与利用 Protel 所画的原理图有本质的区别）与传统的器件连接方式完全一样，很容易被人接受，而且编辑器中有许多现成的单元器件可以利用，自己也可以根据需要设计元件。然而原理图输入法的优点同时也是它的缺点：① 随着设计规模增大，设计的易读性迅速下降，对于图中密密麻麻的电路连线，极难搞清电路的实际功能；② 一旦完成，电路结构的改变将十分困难，因而几乎没有可再利用的设计模块；③ 移植困难、入档困难、交流困难、设计交付困难，因为不可能存在一个标准化的原理图编辑器。

（2）状态图输入方式：以图形的方式表示状态图进行输入。填好时钟信号名、状态转换条

件、状态机类型等要素后，就可自动生成 VHDL 程序。这种设计方式简化了状态机的设计，比较流行。

（3）VHDL 软件程序的文本方式：最一般化、最具普遍性的输入方法，任何支持 VHDL 的 EDA 工具都支持文本方式的编辑和编译。

2．逻辑综合和优化

要把 VHDL 的软件设计与硬件的可实现挂钩，需要利用 EDA 软件系统的综合器进行逻辑综合。

综合器的功能是将设计者在 EDA 平台上为某个系统项目完成的 HDL、原理图或状态图形的描述，针对给定硬件结构组件进行编译、优化、转换和综合，最终获得门级电路，甚至更底层的电路描述文件。由此可见，综合器工作前，必须给定最后实现的硬件结构参数，它的功能就是将软件描述与给定硬件结构用某种网表文件的方式关联起来。显然，综合器是软件描述与硬件实现的桥梁。综合过程就是将电路的高级语言描述转换成低级语言描述的、可与 FPGA/CPLD 或构成 ASIC 的门阵列基本结构相映射的网表文件。

由于 VHDL 仿真器的行为，仿真功能是面向高层次的系统仿真，只能对 VHDL 的系统描述做可行性的评估测试，而不针对任何硬件系统，因此基于这一仿真层次的许多 VHDL 语句不能被综合器所接受。也就是说，这类语句的描述无法在硬件系统中实现（至少是现阶段），这时综合器不支持的语句在综合过程中将被忽略。综合器对源 VHDL 文件的综合是针对某一 PLD 供应商的产品系列的，因此综合后的结果可以为硬件系统所接受，具有硬件可实现性。

3．目标器件的布线/适配

逻辑综合通过后，必须利用适配器将综合后的网表文件针对某一具体的目标器件进行逻辑映射操作，其中包括底层器件配置、逻辑分割、逻辑优化、布线与操作，适配完成后可以利用适配所产生的仿真文件做精确的时序仿真。

适配器的功能是将由综合器产生的网表文件配置到指定的目标器件中，产生最终的下载文件，如 JEDEC 格式的文件。适配所选定的目标器件（FPGA/CPLD 芯片）必须属于原综合器指定的目标器件系列。对于普通可编程模拟器件所对应的 EDA 软件来说，一般仅需包含一个适配器，如 Lattice 的 PAC-DESIGNER。通常，EDA 软件中的综合器可由专业的第三方 EDA 公司提供，而适配器则需由 FPGA/CPLD 供应商自己提供，因为适配器的适配对象直接与器件结构对应。

4．目标器件的编程/下载

若编译、综合、布线/适配和行为仿真、功能仿真、时序仿真等过程都未发现问题，即满足原设计的要求，则可将由 FPGA/CPLD 布线/适配器产生的配置/下载文件，通过编程器或下载电缆载入目标芯片 FPGA 或 CPLD 中。

5．设计过程中的有关仿真

在综合以前，可以先对 VHDL 所描述的内容进行行为仿真，即将 VHDL 设计源程序直接送到 VHDL 仿真器中仿真，这就是所谓的 VHDL 行为仿真。因为此时的仿真只是根据 VHDL 的语义进行的，与具体电路无关。在这时的仿真中，可以充分发挥 VHDL 中的适用于仿真控制的语句及有关的预定义函数和库文件。

在综合之后，VHDL 综合器一般都可生成一个 VHDL 网表文件。网表文件中描述的电路与生成的 EDIF/XNF 等网表文件一致。VHDL 网表文件采用 VHDL 语法，只是其中的电路描述采用了结构描述方法，即首先描述了最基本的门电路，然后将这些门电路用例化语句连接起来。这样的 VHDL 网表文件再送到 VHDL 仿真器中进行所谓的功能仿真，仿真结果与门级仿真器所进行的功能仿真的结果基本一致。

需要注意的是仿真器有两种：一种是 VHDL 仿真器，另一种是门级仿真器，它们都能进行功能仿真和时序仿真。所不同的是，仿真用的文件格式不同，即网表文件不同。这里所说的网表（Netlist）特指电路网络，网表文件描述了一个电路网络。目前流行多种网表文件格式，其中最通用的是 EDIF 格式的网表文件，Xilinx 公司的 XNF 网表文件格式也很流行，不过一般只在使用 Xilinx 的 FPGA/CPLD 时才会用到 XNF 格式。VHDL 文件格式也可用来描述电路网络，即采用 VHDL 语法描述各级电路互连，称之为 VHDL 网表。

功能仿真仅对 VHDL 描述的逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，仿真过程不涉及具体器件的硬件特性，如延时特性。时序仿真是接近真实器件运行的仿真，仿真过程中已将器件特性考虑进去，因而仿真精度要高得多。但是，时序仿真的仿真文件必须来自针对具体器件的布线/适配器所产生的仿真文件。综合后所得的 EDIF/XNF 门级网表文件通常作为 FPGA 布线器或 CPLD 适配器的输入文件。通过布线/适配处理后，布线/适配器将生成一个 VHDL 网表文件，这个网表文件中包含了较为精确的延时信息，网表文件中描述的电路结构与布线/适配后的结果是一致的。此时，将这个 VHDL 网表文件送到 VHDL 仿真器中进行仿真，即可得到精确的时序仿真结果。

6．硬件仿真/硬件测试

这里所说的硬件仿真是针对 ASIC 设计而言的。在 ASIC 设计中，比较常用的方法是利用 FPGA 对系统的设计进行功能检测，通过后再将其 VHDL 设计以 ASIC 形式实现；而硬件测试则是针对 FPGA 或 CPLD 直接用于应用系统的检测而言的。

硬件仿真和硬件测试的目的，是为了在更真实的环境中检验 VHDL 设计的运行情况，特别是对于 VHDL 程序设计上不十分规范、语义上含有一定歧义的程序。一般的仿真器包括 VHDL 行为仿真器和 VHDL 功能仿真器，它们对于同一 VHDL 设计的“理解”，即仿真模型的产生，与 VHDL 综合器的“理解”，即综合模型的产生，常常是不一致的。此外，由于目标器件功能的可行性约束，综合器对于设计的“理解”常在一个有限范围内选择，而 VHDL 仿真器的“理解”是纯软件行为，其“理解”的选择范围要宽得多，结果这种“理解”的偏差势必导致仿真结果与综合后实现的硬件电路在功能上的不一致。当然，还有许多其他的因素也会产生这种不一致。由此可见，VHDL 设计的硬件仿真和硬件测试是十分必要的。

1.7　数字系统的设计

1.7.1　数字系统的设计模型

数字系统指的是交互式的、以离散形式表示的具有存储、传输、信息处理能力的逻辑子系统的集合。用于描述数字系统的模型有多种，各种模型描述数字系统的侧重点不同。下面介绍一种普遍采用的模型。这种模型根据数字系统的定义，将整个系统划分为两个模块或两个子系

统：数据处理子系统和控制子系统。

数据处理子系统主要完成数据的采集、存储、运算和传输。数据处理子系统主要由存储器、运算器、数据选择器等功能电路组成。数据处理子系统与外界进行数据交换，在控制子系统（或称控制器）发出的控制信号作用下，数据处理子系统将进行数据的存储和运算等操作。数据处理子系统将接收由控制器发出的控制信号，同时将自己的操作进程或操作结果作为条件信号传送给控制器。

控制子系统是执行数字系统算法的核心，具有记忆功能，因此控制子系统是时序系统。控制子系统由组合逻辑电路和触发器组成，与数据处理子系统共用时钟。控制子系统的输入信号是外部控制信号和由数据处理子系统送来的条件信号，按照数字系统设计方案要求的算法流程，在时钟信号的控制下进行状态的转换，同时产生与状态和条件信号相对应的输出信号，该输出信号将控制数据处理子系统的具体操作。

把数字系统划分成数据处理子系统和控制子系统进行设计，只是一种手段而非目的。它用来帮助设计者有层次地理解和处理问题，进而获得清晰、完整、正确的电路图。因此，数字系统的划分应当遵循自然、易于理解的原则。

设计一个数字系统时，采用这种模型的优点如下。

（1）把数字系统划分为控制子系统和数据处理子系统两个主要部分，使设计者面对的电路规模减小，二者可以分别设计。

（2）由于数字系统中控制子系统的逻辑关系比较复杂，将其独立划分出来后，可突出设计重点和分散设计难点。

（3）当数字系统划分为控制子系统和数据处理子系统后，逻辑分工清楚，各自的任务明确，因此可以使电路的设计、调测和故障处理都比较方便。

但采用该模型设计一个数字系统时，必须先分析和找出实现系统逻辑的算法，根据具体的算法要求提出系统内部的结构要求，再根据各个部分分担的任务，划分出控制子系统和数据处理子系统。算法不同，系统的内部结构不同，控制子系统和数据处理子系统电路也不同。有时控制子系统和数据处理子系统的界限划分比较困难，需要反复比较和调整才能确定。

1.7.2 数字系统的设计方法

数字系统设计有多种方法，如模块设计法、自顶向下设计法和自底向上设计法等。

数字系统的设计一般采用自顶向下、由粗到细、逐步求精的方法。自顶向下是指将数字系统的整体逐步分解为各个子系统和模块，若子系统规模较大，则还需将子系统进一步分解为更小的子系统和模块，层层分解，直至整个系统中各子系统关系合理，并便于逻辑电路级的设计和实现为止。采用该方法设计时，高层设计进行功能和接口描述，说明模块的功能和接口，模块功能的详细描述在下一设计层次说明，底层的设计才涉及具体的寄存器和逻辑门电路等实现方式的描述。

采用自顶向下设计方法的优点如下。

（1）自顶向下设计方法是一种模块化设计方法。它对设计的描述从上到下，逐步由粗略到详细，符合常规的逻辑思维习惯。由于高层设计与器件无关，设计易于在各种集成电路工艺或可编程器件之间移植。

（2）适合多个设计者同时进行设计。随着技术的不断进步，许多设计由一个设计者已无法

完成，而必须经过多个设计者分工协作来完成。在这种情况下，应用自顶向下的设计方法便于多个设计者同时进行设计，对设计任务进行合理分配，并用系统工程的方法对设计进行管理。

针对具体的设计，实施自顶向下的设计方法的形式会有所不同，但均需遵循逐层分解功能、分层次进行设计的原则。同时，应在各个设计层次上，考虑相应的仿真验证问题。

1.7.3　数字系统的设计准则

进行数字系统设计时，通常需要考虑多方面的条件和要求，如设计的功能和性能要求，元器件的资源分配和设计工具的可实现性，系统的开发费用和成本等。虽然具体设计的条件和要求千差万别，实现的方法也各不相同，但数字系统设计还是具备一些共同的方法和准则。

1．分割准则

自顶向下的设计方法或其他层次化的设计方法，需要对系统功能进行分割，然后用逻辑语言进行描述。分割过程中，若分割过粗，则不易用逻辑语言表达；分割过细，则带来不必要的重复和烦琐。因此，分割的粗细需要根据具体的设计和设计工具情况而定。要掌握分割程度，可遵循以下的原则：分割后底层的逻辑块应适合用逻辑语言进行表达；相似的功能应设计成共享的基本模块；接口信号尽可能少；同层次的模块之间，在资源和 I/O 分配上，尽可能平衡，以使结构匀称；模快的划分和设计，尽可能做到通用性好、易于移植。

2．系统的可观测性

在系统设计中，应同时考虑功能检查和性能的测试，即系统观测性的问题。一些有经验的设计者会自觉地在设计系统的同时，设计观测电路，即观测器，指示系统内部的工作状态。

要建立观测器，应遵循以下原则：具有系统的关键点信号，如时钟、同步信号和状态等信号；具有代表性的节点和线路上的信号；具备“系统工作是否正常”的简单判断能力。

3．同步和异步电路

异步电路会造成较大延时和逻辑竞争，容易引起系统的不稳定，而同步电路则按照统一的时钟工作，稳定性好。因此在设计时，应尽可能采用同步电路进行设计，避免使用异步电路。在必须使用异步电路时，应采取措施来避免竞争和增加稳定性。

4．最优化设计

由于可编程器件的逻辑资源、连接资源和 I/O 资源有限，器件的速度和性能也是有限的，用器件设计系统的过程相当于求最优解的过程。因此，需要给定两个约束条件：边界条件和最优化目标。

所谓边界条件，是指器件的资源及性能限制。最优化目标有多种，设计中常见的最优化目标有：器件资源利用率最高；系统工作速度最快，即延时最小；布线最容易，即可实现性最好。具体设计中，各个最优化目标间可能会产生冲突，这时应满足设计的主要要求。

5．系统设计的艺术

一个系统的设计，通常需要经过反复的修改、优化才能达到设计的要求。一个好的设计，应该满足“和谐”的基本特征，对数字系统可以根据以下几点做出判断：设计是否总体上流畅，无拖泥带水的感觉；资源分配、I/O 分配是否合理，是否没有任何设计上和性能上的瓶颈，系

统结构是否协调；是否具有良好的可观测性；是否易于修改和移植；器件的特点是否能得到充分发挥。

1.7.4 数字系统的设计步骤

1．系统任务分析

数字系统设计中的第一步是明确系统的任务。在设计任务书中，可用各种方式提出对整个数字系统的逻辑要求，常用的方式有自然语言、逻辑流程图、时序图或几种方法的结合。当系统较大或逻辑关系较复杂时，系统任务（逻辑要求）逻辑的表述和理解都不是一件容易的工作。所以，分析系统的任务必须细致、全面，不能有理解上的偏差和疏漏。

2．确定逻辑算法

实现系统逻辑运算的方法称为逻辑算法，简称算法。一个数字系统的逻辑运算往往有多种算法，设计者的任务不仅是找出各种算法，还必须比较优劣，取长补短，从中确定最合理的一种。数字系统的算法是逻辑设计的基础，算法不同，系统的结构也不同，因此算法的合理与否直接影响系统结构的合理性。确定算法是数字系统设计中最具创造性的一环，也是最难的一步。

3．建立系统及子系统模型

算法明确后，应根据算法构造系统的硬件框架（也称系统框图），将系统划分为若干部分，各部分分别承担算法中不同的逻辑操作功能。如果某一部分的规模仍较大，则需进一步划分。划分后的各部分应逻辑功能清楚，规模大小合适，便于进行电路级的设计。

4．系统（或模块）逻辑描述

当系统中各个子系统（指最低层子系统）和模块的逻辑功能与结构确定后，则需采用比较规范的形式来描述系统的逻辑功能。设计方案的描述方法可以有多种，常用的有方框图、流程图和描述语言等。

对系统的逻辑描述可先采用较粗略的逻辑流程图，再将逻辑流程图逐步细化为详细逻辑流程图，最后将详细逻辑流程图表示成与硬件有对应关系的形式，为下一步的电路级设计提供依据。

5．逻辑电路级设计及系统仿真

电路级设计是指选择合理的器件和连接关系来实现系统逻辑要求。电路级设计的结果常采用两种方式来表达：电路图方式和硬件描述语言方式。EDA 软件允许以这两种方式输入，以便做后续的处理。

电路设计完成后，必须验证设计是否正确。早期，人们只能通过搭试硬件电路才能得到设计的结果。今天，数字电路设计的 EDA 软件都具有仿真功能，因此可先进行系统仿真，系统仿真结果正确后再进行实际电路的测试。由于 EDA 软件的验证结果十分接近实际结果，因此可极大地提高电路设计的效率。

6．系统的物理实现

物理实现是指用实际的器件实现数字系统的设计，用仪表测量设计的电路是否符合设计要求。今天的数字系统往往采用大规模和超大规模集成电路，由于器件集成度高、导线密集，故一般在电路设计完成后即设计印制电路板，在印制电路板上组装电路进行测试。需要注意的是，印制电路板本身的物理特性也会影响电路的逻辑关系。

习　题

1.1　EDA 的中文含义是什么？
1.2　什么是 EDA 技术？
1.3　利用 EDA 技术进行电子系统设计有什么特点？
1.4　什么叫可编程逻辑器件（PLD）？
1.5　自顶向下的设计方法有何优点？