

情景 1 概 述

1.1 EDA 技术及其发展

EDA (Electronic Design Automation) 即电子设计自动化。EDA 技术,就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,通过相关的开发软件,自动完成用软件方式设计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 技术进行电子系统的设计,具有以下特点: ① 用软件的方式设计硬件; ② 用软件方式设计的电子系统到硬件系统的转换是由相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。

EDA 技术发展有以下三个阶段:

1. 20 世纪 70 年代  MOS 工艺  CAD 概念

20 世纪 70 年代，MOS 工艺在集成电路制作方面得到广泛应用，可编程逻辑技术及器件已经出现。计算机在科研领域的广泛应用，促使了 CAD 技术的出现。CAD (Computer Assist Design) 即计算机辅助设计。在这一阶段，人们开始利用计算机取代手工劳动，辅助进行集成电路版图设计、PCB 布局布线等工作。

2. 20 世纪 80 年代  CMOS 时代  出现 FPGA

20 世纪 80 年代，集成电路设计进入 CMOS 时代，复杂可编程逻辑器件 (CPLD) 已经进入商业应用，80 年代末，出现了 FPGA。CAD 技术和 CAE 技术应用更加广泛。CAE (Computer Assist Engineering Design) 即计算机辅助工程设计，它集逻辑图输入、逻辑模拟、测试码生成、电路模拟、版图设计、版图验证等工具于一体，构成一个较完整的 IC 设计系统。在这一阶段，还出现了各种硬件描述语言。

3. 20 世纪 90 年代  ASIC 设计技术  EDA 技术

20 世纪 90 年代，随着硬件描述语言的标准化得到进一步的确立，HDL 输入取代了逻辑输入，计算机辅助工程、辅助分析和辅助设计在电子技术领域获得更加广泛的应用。集成电路设计工艺步入了超深亚微米阶段，百万门以上的大规模可编程逻辑器件的陆续面世，以及基于计算机技术的、面向用户的低成本大规模 ASIC 设计技术的应用，促进了 EDA 技术的形成。

EDA 技术在进入 21 世纪后，得到了更大的发展：

- ① EDA 使得电子领域各学科的界限更加模糊，更加互为包容；
- ② 更大规模的 FPGA 和 CPLD 器件不断推出；
- ③ 基于 EDA 工具用于 ASIC 设计的标准单元包括大规模电子系统及复杂 IP 核模块；
- ④ 软硬件 IP（Intellectual Property）核在电子行业得到广泛应用；
- ⑤ SoC 高效低成本设计技术变得成熟；
- ⑥ 系统级硬件描述语言出现（如 System C）使复杂电子系统设计和验证趋于简单。

1.2 EDA 技术实现目标

1.2.1 目标

EDA 技术实现目标是完成专用集成电路 ASIC 的设计和实现，如图 1.1 所示。

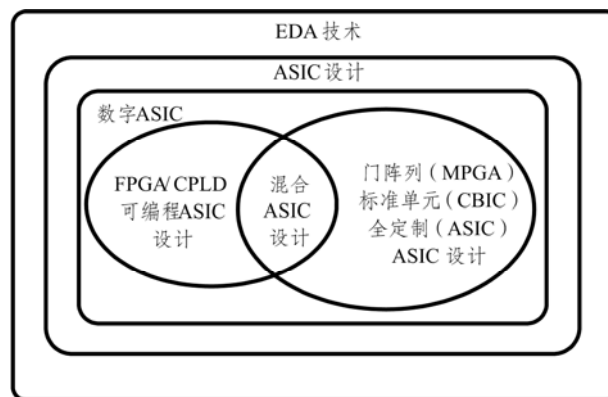


图 1.1 EDA 技术实现目标

ASIC（Application Specific Integrated Circuits）是指应特定用户要求或特定应用需要而设计制造的集成电路。

ASIC 的概念早在 20 世纪 60 年代就有人提出，但其真正发展是在进入 20 世纪 80

年代以后。其技术特点是工艺和设计技术均已成熟，由于电子产品竞争激烈，迫使厂商采用 ASIC 取代中小规模 IC 构成系统。

采用 ASIC 来实现系统集成具有如下优点：① 缩小体积，减轻重量，降低功耗；② 提高可靠性；③ 易于获得高性能；④ 可增强保密性；⑤ 大批量应用时可降低系统成本。

与通用 IC 相比，ASIC 具有如下特点：① 功能强、品种多、批量小；② 使用寿命与整机的寿命有关。

1.2.2 ASIC 技术发展对当代电子系统设计的影响

ASIC 技术发展对当代电子系统设计的影响主要体现在两个方面：① 用 ASIC 实现系统集成；② 系统和电路工程师参与 ASIC 设计。

过去，电子系统设计的基本思路是：选用中小规模的通用标准 IC 构成电路、子系统、系统。采用“Bottom.up”设计方法。这样设计出的电子系统，所用元件的种类和数量均较多，体积功耗大，可靠性差，且调试困难。

现在的电子系统设计采用 Top.down 设计思路：由整机单位对整个系统进行方案设计、功能划分，系统的关键电路用一片或几片 ASIC 实现。且这些 ASIC 是由系统或电路设计师亲自参与设计的，完成电路到芯片版图的设计后，再交由 IC 工厂投片加工，或是用可编程专用集成电路（例如 FPGA）现场编程实现。

在新形势下，作为电子设计工程师，我们担当的角色发生了某种程度的变化。

过去——我们仅仅是 IC 用户，现在——要参与到 ASIC 的设计与开发中去。这就要求我们除了要有线路和系统的基础外，还要了解集成电路的内部结构、生产工艺、设计原则和设计方法等方面的知识。

1.2.3 ASIC 的实现途径

ASIC 的实现可通过三种途径来完成：

1. 超大规模可编程逻辑器件

FPGA 和 CPLD 是实现这一途径的主流器件，它们的特点是直接面向用户，具有极大的灵活性和通用性，使用方便，硬件测试和实现快捷，开发效率高，成本低，上市时间短，技术维护简单，工作可靠性好等。由于 FPGA 和 CPLD 的开发工具、开发流程和使用方法与 ASIC 有类似之处，因此，这类器件通常也被称为可编程 ASIC。

2. 半定制或全定制 ASIC

基于 EDA 技术的半定制或全定制 ASIC，根据它们的实现工艺，可统称为掩模 ASIC。其特点如图 1.2 所示。可编程 ASIC 与掩模 ASIC 相比，其不同之处就在于它具有面向用户的灵活多样的可编程性。

1) 门阵列 ASIC

门阵列芯片包括预定制的相连的 PMOS 和 NMOS 晶体管行。在设计中，用户可以借助 EDA 工具将原理图或硬件描述语言模型映射为相应门阵列晶体管配置，创建一个指定金属互连路径文件，从而完成门阵列 ASIC 的开发。现在，Altera 公司的 HardCopy 技术，可以提供一种把 FPGA 的设计转化为门阵列 ASIC 的途径。



图 1.2 掩模 ASIC 特点

2) 标准单元 ASIC

目前大多数 ASIC 是基于标准单元库进行设计的 (Cell Based Integrated Circuits, CBIC)。库中包括不同复杂性的逻辑元件: SSI 逻辑块、MSI 逻辑块、数据通道模块、存储器、IP 乃至系统级模块。库中每个单元的版图已事先设计好, 并已经过工艺和性能验证, 使用者只需利用 EDA 软件使用各模块即可, 而不必去了解各电路的细节。

3) 全定制 ASIC

在针对特定工艺建立的设计规则下, 全定制 ASIC 的设计者对电路的设计有完全的控制权, 设计者可以使用版图编辑工具, 对每个晶体管的版图尺寸、位置及互连线进行设计。该领域的一个例外是混合信号设计, 使用通信电路中的 ASIC 可以定制设计其模拟部分。

3. 混合 ASIC

混合 ASIC (不是指数摸混合 ASIC) 主要指既具有面向用户的 FPGA 可编程功能和逻辑资源, 同时也含有可方便调用和配置的硬件标准单元模块, 如 CPU、RAM、ROM、硬件加法器、硬件乘法器、锁相环等模块。Xilinx、Altera 等公司已经推出了这方面的器件, 如 Virtex.4 系列和 Stratix II 系列等。混合 ASIC 已成为 SOC 和 SOPC 设计的便捷途径。

1.3 硬件描述语言 VHDL

硬件描述语言是 EDA 技术的重要组成部分。常见的硬件描述语言有：VHDL、Verilog HDL、System Verilog、System C。其中 VHDL、Verilog 使用最普遍，也拥有几乎所有主流 EDA 工具的支持。而 System Verilog 和 System C 这两种硬件描述语言主要是针对系统级的设计，目前还处于不断完善的过程中。

标准硬件描述语言 VHDL (Very High Speed Integrated Circuit Hardware Description Language) 进行工程设计的优点是多方面的。

(1) 与其他的硬件描述语言相比，VHDL 具有更强的行为描述能力，从而决定了它成为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构、从逻辑行为上描述和设计大规模电子系统的重要保证。

(2) VHDL 丰富的仿真语句和库函数，使得在任何大系统的设计早期就能查验设计系统的功能可行性，随时可对设计系统进行仿真模拟。

(3) VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。如果要高速、高效地完成符合市场需求的大规模系统，必须有多人甚至多个开发组共同并行工作才能实现。

(4) 对于用 VHDL 完成的一个确定的设计，可以利用 EDA 工具进行逻辑综合和优化，并自动把 VHDL 描述设计转变成门级网表。

(5) VHDL 对设计的描述具有相对独立性，设计者可以不懂硬件的结构，也不必

考虑最终设计实现的目标器件是什么，只要进行独立的设计即可。

1.4 VHDL 综合

综合就是把某些东西结合到一起，把设计抽象层次的一种表示转化成另一种表示的过程。在电子设计领域中综合的概念可以表示为：将用行为和功能层次表达的电子系统转换成低层次的、便于具体实现的模块组合装配的过程。

如图 1.3 所示，VHDL 硬件描述语言综合器可以将抽象的 VHDL 描述转化成低层次的门级网表；而软件程序编译器可以将高级语言程序转化成低级的机器代码，二者本质上相同吗？

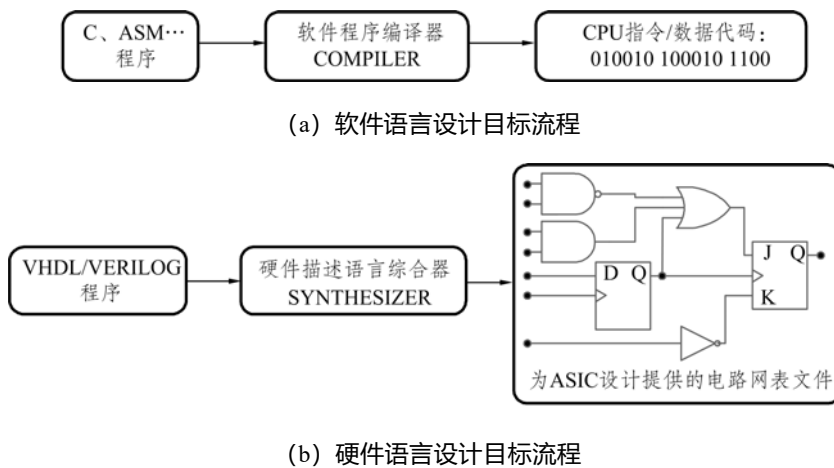


图 1.3 设计目标流程

它们的本质是不同的。

编译器将软件程序翻译成基于某种特定 CPU 的机器代码，该代码仅限于这种 CPU 而不能移植，并且机器代码不代表硬件结构，更不能改变 CPU 的硬件结构，只能被 CPU

利用。如果脱离了已有的硬件环境（CPU），机器代码将失去意义。

综合器可以将抽象的 VHDL 描述转化成底层的电路结构门级网表文件，这种网表文件不依赖于任何特定的硬件环境，因此可以独立存在，并且能轻易地被移植到任何通用的硬件环境中，如 ASIC、FPGA 等。VHDL 综合器运行流程如图 1.4 所示。

综合器在接受 VHDL 程序并准备对其综合前，必须获得与最终实现电路硬件特征相关的工艺库的信息，以及获得优化综合的各种约束条件。一般约束条件可以分为三种：设计规则、时间约束、面积约束。

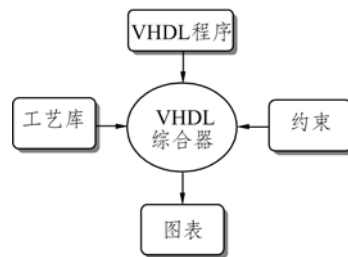


图 1.4 VHDL 综合器运行流程

综合器在将硬件描述语言表达的电路功能转化成具体的电路结构网表的过程中，具有明显的能动性和创造性，它不是机械地一一对应地翻译，而是根据设计库、工艺库以及预先设置的各类约束条件，选择最优的方式完成对电路结构的设计。

另外，并不是所有的 VHDL 语法都是可综合的，不同的综合器所支持的 VHDL 子集也不相同。因此，相同的 VHDL 源代码、不同的 VHDL 综合器可能综合出结构和功能并不完全相同的电路系统。

1.5 基于 VHDL 的自顶向下设计方法

基于 VHDL 的自顶向下的设计流程如图 1.5 所示，主要包括设计说明书、建立 VHDL 行为模型、VHDL 行为仿真、VHDL.RTL 级建模、前端功能仿真、逻辑综合、测试向量生成、测试向量生成、功能仿真、结构综合、门级时序仿真、硬件测试等内容。

(1) 提出设计说明书。就是用自然语言提出系统项目的功能特点和技术参数等。

(2) 建立 VHDL 行为模型。就是将设计说明书转化为 VHDL 行为模型。在这个过程中可以使用 VHDL 的所有语句而不必考虑其可综合性。这一建模行为的目的是通过 VHDL 仿真器对整个系统进行系统行为仿真和性能评估。

(3) VHDL 行为仿真。这一阶段可以利用 VHDL 仿真器对顶层系统的行为模型进行仿真测试，检查模拟结果，继而进行修改和完善。这一过程与最终实现的硬件无关。

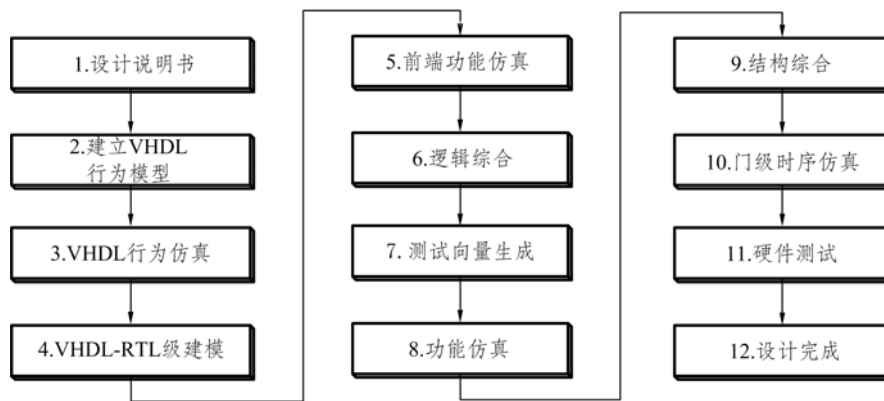


图 1.5 自顶向下的设计流程

(4) VHDL.RTL 级建模。VHDL 语法中所有语句都可以进行仿真，但只有一部分可以综合成门级网表，因此在这一阶段，必须将行为模型转化为可综合的 VHDL.RTL 级模型。

(5) 前端功能仿真。对 VHDL.RTL 级模型进行仿真，称为功能仿真。尽管 VHDL.RTL