

第 1 章 EDA 技术概述

我们已经进入数字化和信息化的时代，其特点是各种数字产品广泛应用。现代数字产品在性能提高、复杂度增大的同时，更新换代的步伐也越来越快，实现这种进步的因素在于芯片制造技术和设计技术的进步。

芯片制造技术以微细加工技术为代表，目前已进展到深亚微米阶段，可以在几平方厘米的芯片上集成数千万个晶体管。摩尔曾经对半导体集成技术的发展做出预言：大约每 18 个月，芯片的集成度提高为原来的两倍，功耗下降一半，他的预言被人们称为摩尔定律（Moore's Law）。几十年来，集成电路的发展与这个预言非常吻合，数字器件经历了从 SSI、MSI、LSI 到 VLSI，直到现在的 SoC（System on Chip，芯片系统），我们已经能够把一个完整的电子系统集成在一个芯片上。还有一种器件的出现极大改变了设计制作电子系统的方式与方法，这就是可编程逻辑器件（Programmable Logic Device，PLD）。PLD 器件是 20 世纪 70 年代后期发展起来的一种器件，它经历了可编程逻辑阵列（Programmable Logic Array，PLA）、通用阵列逻辑（Generic Array Logic，GAL）等简单形式到现场可编程门阵列（Field Programmable Gate Array，FPGA）和复杂可编程逻辑器件（Complex Programmable Logic Device，CPLD）的高级形式的发展，它的广泛使用不仅简化了电路设计，降低了研制成本，提高了系统可靠性，而且给数字系统的整个设计和实现过程带来了革命性的变化。

电子系统的设计理念和设计方法也发生了深刻的变化，从电子 CAD（Computer Aided Design）、电子 CAE（Computer Aided Engineering）到电子设计自动化（Electronic Design Automation，EDA），设计的自动化程度越来越高，设计的复杂性也越来越强。

EDA 技术已成为现代电子设计技术的有力工具，没有 EDA 技术的支持，要完成超大规模集成电路的设计和制造是不可想象的，反过来，生产制造技术的进步又不断对 EDA 技术提出新的要求，促使其不断向前发展。

1.1 EDA 技术及其发展

在现代数字系统的设计中，EDA 技术已经成为一种普遍的工具。对设计者而言，熟练地掌握 EDA 技术，可以极大地提高工作效率，起到事半功倍的效果。

EDA（电子设计自动化）技术没有一个精确的定义，我们可以这样来认识，所谓的 EDA 技术就是以计算机为工具，设计者基于 EDA 软件平台，采用原理图或硬件描述语言（HDL）完成设计输入，然后由计算机自动完成逻辑综合、优化、布局布线和仿真，直至完成目标芯片（CPLD、FPGA）的适配和编程下载等工作（甚至是完成 ASIC 专用集成电路掩膜设计），上述辅助进行电子设计的软件工具及技术统称为 EDA。EDA 技术的发展以计算机科学、微电子技术的发展为基础，并融合了应用电子技术、智能技术，以及计算机图形学、拓扑学、计算数学等众多学科的最新成果。EDA 技术经历了一个由简单到复杂、由初级到高级不断发展进步的过程。从 20 世纪 70 年代开始，人们就已经开始基于计算机开发出一些软件工具帮助设计者完成电路系统的设计任务，以代替传统的手工设计方法，随着计算机软件和硬件技术水平的提高，EDA 技术也在不断进步，大致经历了下面三个发展阶段。

1. CAD 阶段

电子 CAD 阶段是 EDA 技术发展的早期阶段 (时间大致为 20 世纪 70 年代至 80 年代初)。在这个阶段,一方面,计算机的功能还比较有限,个人计算机还没有普及;另一方面,电子设计软件的功能也较弱。人们主要是借助计算机对所设计电路的性能进行模拟和预测;另外,就是完成 PCB 的布局布线、简单版图的绘制等工作。

2. CAE 阶段

集成电路规模的扩大,电子系统设计的逐步复杂,使得电子 CAD 的工具逐步完善和发展,尤其是人们在设计方法学、设计工具集成化方面取得了长足的进步,EDA 技术就进入了电子 CAE 阶段 (时间大致为 20 世纪 80 年代初至 90 年代初)。在这个阶段,各种单点设计工具、各种设计单元库逐渐完备,并且开始将许多单点工具集成在一起使用,大大提高了工作效率。

3. EDA 阶段

20 世纪 90 年代以来,微电子工艺有了显著发展,工艺水平已经达到了深亚微米级,在一个芯片上已经可以集成上千万乃至上亿的晶体管,芯片的工作速度达到了 Gbps 级,这样就对电子设计工具提出了更高的要求,也促进了设计工具的发展。

在今天,EDA 技术已经成为电子设计的普遍工具,无论是设计集成电路还是设计普通的电子电路,没有 EDA 工具的支持,都是难以完成的。EDA 技术的使用包括电子工程师进行电子系统开发的全过程,以及进行开发设计涉及的各个方面。从一个角度来看,EDA 技术可粗略分为系统级、寄存器传输级 (RTL)、门级和版图级几个层次的辅助设计过程;从另一个角度来看,EDA 技术包括电子电路设计的各个领域,即从低频电路到高频电路、从线性电路到非线性电路、从模拟电路到数字电路、从 PCB 设计到 FPGA 开发等,EDA 技术的功能和范畴如图 1.1 所示。

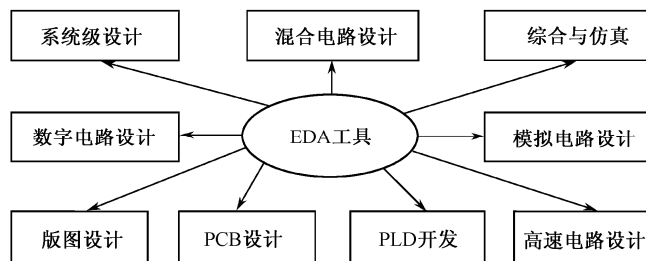


图 1.1 EDA 技术的功能和范畴

进入 21 世纪后,EDA 技术得到更快的发展,开始步入一个新的时期,突出地表现在以下几个方面。

(1) 电子技术各个领域全方位融入 EDA 技术,除日益成熟的数字技术外,可编程模拟器件的设计技术也有了很大的进步。EDA 技术使得电子领域各学科的界限更加模糊、相互包容和渗透,如模拟与数字、软件与硬件、系统与器件、ASIC 与 FPGA、行为与结构等,软硬件协同设计技术也成为 EDA 技术的一个发展方向。

(2) IP (Intellectual Property) 核在电子设计领域得到更广泛的应用,进一步缩短设计周期,提高设计效率。基于 IP 核的 SoC 设计技术趋向成熟,电子设计成果的可重用性得到提高。

(3) 嵌入式微处理器软核的出现,更大规模的 FPGA/CPLD 器件的不断推出,使得 SOPC (System On Programmable Chip, 可编程芯片系统) 步入实用化阶段,在一片 FPGA 芯片中实现一个完备的系

统成为可能。

(4) 用 FPGA (Field Programmable Gate Array, 现场可编程门阵列) 器件实现完全硬件的 DSP (数字信号处理) 处理成为可能, 用纯数字逻辑进行 DSP 模块的设计, 为高速数字信号处理算法提供了实现途径, 并有力地推动了软件无线电技术的实用化。

(5) 在设计和仿真两方面支持标准硬件描述语言的 EDA 软件不断推出, 系统级、行为验证级硬件描述语言的出现 (如 System C) 使得复杂电子系统的设计和验证更加高效。在一些大型的系统设计中, 设计验证工作非常艰巨, 这些高效的 EDA 工具的出现, 减轻了开发人员的工作量。

除了上述的发展趋势, 现代 EDA 技术和 EDA 工具还呈现出以下一些共同的特点。

(1) 硬件描述语言 (HDL) 标准化程度提高

硬件描述语言 (Hardware Description Language, HDL) 不断进化, 其标准化程度越来越高, 便于设计的复用、交流、保存和修改, 也便于组织大规模、模块化的设计。标准化程度最高的硬件描述语言是 Verilog 和 VHDL, 它们早已成为 IEEE 标准, 并且有新的版本获得通过, 比如 Verilog 有 Verilog-1995 和 Verilog-2001 两个版本, 其功能得到增强, 标准化程度得到提高。

(2) EDA 工具的开放性和标准化程度不断提高

现代 EDA 工具普遍采用标准化和开放性的框架结构, 可以接纳其他厂商的 EDA 工具一起进行设计工作。这样可实现各种 EDA 工具间的优化组合, 并集成在一个易于管理的统一环境之下, 实现资源共享, 有效提高设计者的工作效率, 有利于大规模、有组织地设计开发工作。

EDA 工具已经能接受功能级或 RTL 级 (Register Transport Level) 的 HDL 描述进行逻辑综合和优化。为了能更好地支持自顶向下的设计方法, EDA 工具需要在更高的层级进行综合和优化, 并进一步提高智能化程度, 提高设计的优化程度。

(3) EDA 工具的库 (Library) 更完备

EDA 工具要具有更强大的设计能力和更高的设计效率, 必须配有丰富的库, 比如元件图形符号库、元件模型库、工艺参数库、标准单元库、可复用的宏功能模块库、IP 库等。在电路设计的各个阶段, EDA 系统需要不同层次、不同种类的元件模型库的支持。例如, 原理图输入时需要原理图符号库、宏模块库, 逻辑仿真时需要逻辑单元的功能模型库, 模拟电路仿真时需要模拟器件的模型库, 版图生成时需要适应不同层次和不同工艺的底层版图库等。各种元件模型库的规模和功能是衡量 EDA 工具优劣的一个重要标志。

总而言之, 从之前发展的过程看, EDA 技术一直滞后于制造工艺的发展, 它在制造技术的驱动下不断进步; 从长远看, EDA 技术将随着微电子技术、计算机技术的不断发展而发展。“工欲善其事, 必先利其器”, EDA 工具在现代电子系统的设计中所起的作用越来越大, 未来它将在诸多因素的推动下继续进步。

1.2 Top-down 设计与 IP 核复用

数字系统的设计方法发生了深刻的变化。传统的数字系统采用搭积木的方式进行设计, 即由一些固定功能的器件加上一定的外围电路构成模块, 由这些模块进一步形成各种功能电路, 进而构成系统。构成系统的积木块是各种标准芯片, 如 74/54 系列 (TTL)、4000/4500 系列 (CMOS) 芯片等, 这些芯片的功能是固定的, 用户只能根据需要从这些标准器件中选择, 并按照推荐的电路搭成系统。在设计时几乎没有灵活性可言, 设计一个系统所需的芯片种类多且数量多。

PLD 器件和 EDA 技术的出现改变了这种传统的设计思路, 使人们可以立足于 PLD 芯片来实现各种不同的功能, 新的设计方法能够由设计者自己定义器件的内部逻辑和引脚, 将原来由电路板设计完

成的工作大部分放在芯片的设计中进行。这样不仅可以通过芯片设计实现各种数字逻辑功能，而且由于引脚定义具有灵活性，降低了原理图和印制板设计的工作量及难度，增加了设计的自由度，提高了效率。同时这种设计减少了所需芯片的种类和数量，缩小了体积，降低了功耗，提高了系统的可靠性。

在基于 EDA 技术的设计中，通常有两种设计思路：一种是自顶向下的设计思路，另一种是自底向上的设计思路。

1.2.1 Top-down 设计

Top-down 设计，即自顶向下的设计。这种设计方法首先从系统设计入手，在顶层进行功能方框图的划分和结构设计。在功能级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述，然后用综合工具将设计转化为具体门电路网表，其对应的物理实现可以是 PLD 器件或专用集成电路 (ASIC)。设计的主要仿真和调试过程是在高层次上完成的，这一方面有利于早期发现结构设计上的错误，避免设计工作的浪费，另一方面也减少了逻辑功能仿真的工作量，提高了设计的一次成功率。

在 Top-down 设计中，将设计分成几个不同的层次：系统级、功能级、门级和开关级等，按照自上而下的顺序，在不同的层次上对系统进行设计与仿真。图 1.2 所示为这种设计方式的示意图。由图可见，在 Top-down 的设计过程中，需要有 EDA 工具的支持，有些步骤 EDA 工具可以自动完成，比如综合等，有些步骤 EDA 工具为用户提供了操作平台。Top-down 设计必须经过“设计—验证—修改设计—再验证”的过程，不断反复，直到得到的结果能够完全实现所要求的逻辑功能，并且在速度、功耗、价格和可靠性方面实现较为合理的平衡。不过，这种设计也并非绝对的，在设计的过程中，有时也需要用到自底向上的方法，就是在系统划分和分解的基础上，先进行底层单元设计，然后再逐步向上进行功能块、子系统的设计，直至构成整个系统。

图 1.3 所示为 CPU 的 Top-down 设计方式示意图。首先在系统级划分，将整个 CPU 划分为几个模块，如 ALU、PC、RAM 模块等，对每个模块再分别进行设计与描述，然后通过 EDA 工具将整个设计综合为门级网表，并实现它。在设计过程中，需要进行多次仿真和验证，不断修改设计。

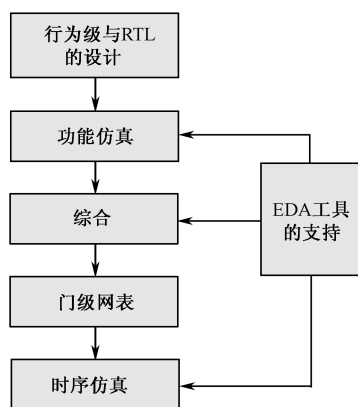


图 1.2 Top-down 设计方式的示意图

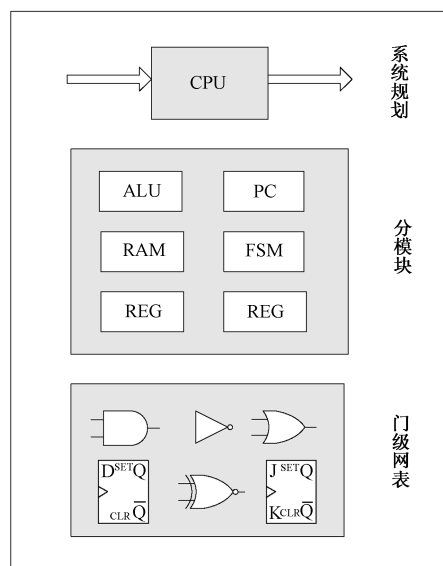


图 1.3 CPU 的 Top-down 设计方式示意图

1.2.2 Bottom-up 设计

Bottom-up 设计，即自底向上的设计，这是一种传统的设计思路。这种设计方式，一般是设计者选择标准集成电路，或者将各种基本单元，如各种门电路及加法器、计数器等模块做成基本单元库，调用这些基本单元，逐级向上组合，直到设计出满足自己需要的系统为止。这样的设计方法就如同用一砖一瓦建造金字塔，不仅效率低、成本高，而且容易出错。

Top-down 设计由于更符合人们逻辑思维的习惯，也容易使设计者对复杂的系统进行合理的划分与不断的优化，因此是目前设计思想的主流。而 Bottom-up 设计往往使设计者关注了细节，而对整个的系统缺乏规划，当设计出现问题时，如果要修改，就会比较麻烦，甚至前功尽弃，不得不从头再来。因此，在数字系统的设计中，主要采用 Top-down 的设计思路，而以 Bottom-up 设计为辅。

1.2.3 IP 复用技术与 SoC

当电子系统的设计越来越向高层发展时，基于 IP 复用 (IP Reuse) 的设计技术越来越显示出优越性。IP (Intellectual Property)，其原来的含义是指知识产权、著作权等，在 IC 设计领域可将其理解为实现某种功能的设计，IP 核 (IP 模块) 则是指完成某种功能的设计模块。

IP 核分为硬核、固核和软核三种类型。软核指的是在寄存器级或门级对电路功能用 HDL 进行描述，表现为 VHDL 或 Verilog 代码。软核与生产工艺无关，不涉及物理实现，给后续设计留很大的空间，增大了 IP 的灵活性和适应性。用户可以对软核的功能加以裁剪以符合特定的应用，也可以对软核的参数进行设置，包括总线宽度、存储器容量、使能或禁止功能块等。硬核指的是以版图形式实现的设计模块，它基于一定的设计工艺，通常用 GDS II 格式表示，不同的客户可以根据自己的需要选用特定生产工艺下的硬核。固核是完成了综合的功能块，通常以网表的形式提交客户使用。软核使用灵活，但其可预测性差，延时不一定能达到要求；硬核可靠性高，能确保性能，如速度、功耗等，能够很快地投入使用。

如图 1.4 所示，由微处理器核 (MPU Core)、数字信号处理器核 (DSP Core)、存储器核 (RAM/ROM)、A/D 核、D/A 核及 USB 接口核等构成一个系统芯片 (SoC)。用户在设计一个系统时，可以自行设计各个功能模块，也可以用 IP 模块来构建。作为设计者来说，想要在短时间内开发出新产品，一个比较好的方法就是使用 IP 核完成设计。目前，还有专门的组织 VSIA (Virtual Socket Interface Association, 虚拟插座接口联盟) 来制定关于 IP 产品的标准与规范。

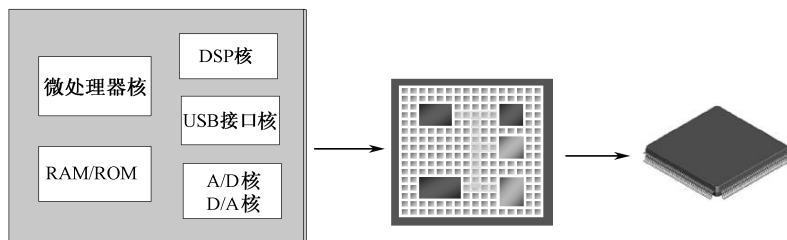


图 1.4 系统芯片 (SoC) 示意图

如上所述，基于 IP 复用的开发给设计者带来了诸多好处，如节省时间、缩短开发周期、避免重复劳动等。当然，IP 的发展还存在一些问题，比如 IP 版权的保护、IP 的保密及 IP 间的集成等。但基于 IP 复用的设计技术无疑会成为电子系统开发的重要手段之一。

系统芯片 (SoC)，或者称为芯片系统、片上系统，是指把一个完整的系统集成在一个芯片上，或

者说是用一个芯片实现一个功能完整的系统。系统芯片可以采用全定制的方式来实现,把设计的网表文件提交给半导体厂家流片就可得到,但采用这种方式风险性高、费用多、周期长。还有一种方式就是采用可编程逻辑器件来实现。CPLD 和 FPGA 的集成度越来越高,速度也越来越快,设计者可以在其上通过编程完成自己的设计。今天,不仅能用它们实现一般的逻辑功能,还可以把微处理器、DSP、存储器、标准接口等功能部件全部集成在其中,真正实现 System on Chip。

微电子制造工艺的进步为 SoC 的实现提供了硬件条件,而 EDA 软件技术的提高则为 SoC 创造了必要的开发平台。目前,EDA 的新工具、新标准和新方法正在向着高层化发展,过去已将设计从晶体管级提高到了逻辑门级,后来,又提高到了寄存器传输级,现在则越来越多地在系统级完成。

在数字系统进入 SoC 时代后,设计方法也随之产生变化。如果把器件的设计视为设计者根据设计规则用软件来搭接已有的不同模块,那么早期的设计是基于晶体管的 (Transistor Based Design)。在这一阶段,设计者最关心的是怎样减小芯片的面积,所以又称为面积驱动的设计 (Area Driving Design,

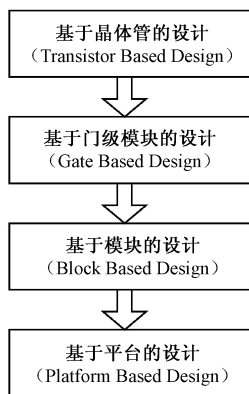


图 1.5 设计方法的演变

ADD)。随着设计方法的改进,出现了以门级模块为基础的设计 (Gate Based Design)。在这一阶段,设计者在考虑芯片面积的同时,更多关注门级模块之间的延时,所以这种设计又称为延时驱动的设计 (Time Driving Design, TDD)。自 20 世纪 90 年代以来,芯片的集成度进一步提高,系统芯片 SoC 的出现使得以 IP 模块复用为基础的设计逐渐流行,这种设计方法称为基于模块的设计方法 (Block Based Design, BBD)。在应用 BBD 方法进行设计的过程中,逐渐产生的一个问题是:在开发完一个产品后,怎么能尽快开发出其系列产品?这样就产生了新的概念——PBD, PBD 是基于平台的设计方法 (Platform Based Design),它是一种基于 IP 的、面向特定应用领域的 SoC 设计环境,可以在更短的时间内设计出满足需要的电路。PBD 的实现依赖如下关键技术的突破:高层次系统级的设计工具、软硬件协同设计技术等。图 1.5 所示为设计方法的演变。

1.3 EDA 设计的流程

EDA 设计的实现主要可选择两类器件,一类是可编程逻辑器件 (PLD),另一类是专用集成电路 (ASIC),这两类器件各有优点。

专用集成电路 (Application Specific Integrated Circuit, ASIC) 是指用全定制方法来实现设计的方式,它在最低层,即物理版图级实现设计,因此也称为掩膜 (Mask) ASIC。采用 ASIC,能得到最高速度、最低功耗和最省面积的设计。它要求设计者必须使用版图编辑工具从晶体管的版图尺寸、位置及连线开始进行设计,以得到芯片的最优性能。在进行版图设计时,设计者需手工设计版图并精心地布局布线,以获得最佳的性能和最小的面积。版图设计完成后,还要进行一系列检查和验证,包括设计规则检查、电学规则检查、连接性检查、版图与电路图一致性检查等,全部通过后,才可以将得到的标准格式的版图文件 (一般为 CIF、GDS II 格式) 交给半导体厂家进行流片。半导体厂家基于母片 (晶圆, Wafer) 通过一系列复杂的工艺制作芯片。ASIC 的设计周期长,设计难度大,制造成本高昂,但可以设计出速度快、功耗低、尽量节省面积芯片,适用于对性能要求很高和批量生产的芯片。

如果想得到设计周期短、投入少、风险小的实现方案,可选择 PLD 器件来实现数字系统。PLD (主要包括 FPGA 和 CPLD) 是一种半定制的器件,器件内已做好各种逻辑资源,用户只需对器件内的资源编程连接就可实现所需要的功能,而且可以反复修改、反复编程,直到满足设计要求。用 PLD 实现

设计直接面向用户，具有其他方法无可比拟的方便性、灵活性和通用性，硬件测试和实现快捷，开发效率高、成本低、风险小。现代 FPGA 器件集成度不断提高，等效门数已达到了千万门级，在器件中，除集成各种逻辑门和寄存器外，还集成了嵌入式块 RAM、硬件乘法器、锁相环、DSP 块等功能模块，使 FPGA 的使用更方便。EDA 开发软件对 PLD 器件也提供了强有力的支持，其功能更全面，兼容性更强。

基于 FPGA/CPLD 器件的数字系统设计流程图如图 1.6 所示，包括设计输入、综合、布局布线、仿真和编程配置等步骤。

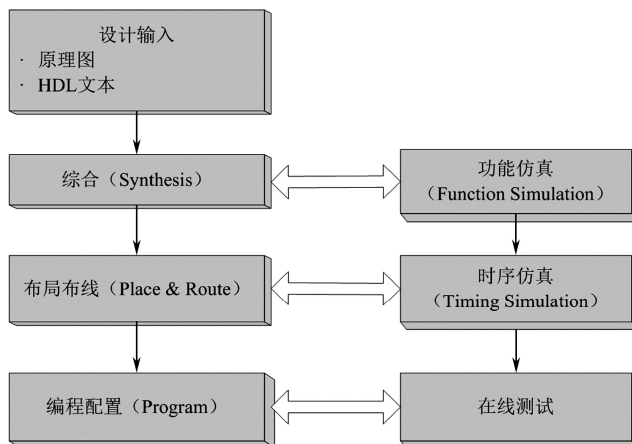


图 1.6 基于 FPGA/CPLD 器件的数字系统设计流程图

1.3.1 设计输入

设计输入 (Design Entry) 是将设计者所设计的电路以开发软件要求的某种形式表达出来，并输入到相应软件中的过程。设计输入有多种表达方式，最常用的是原理图方式和 HDL 文本方式两种。

1. 原理图输入

原理图 (Schematic) 是图形化的表达方式，使用元件符号和连线来描述设计。其特点是适合描述连接关系和接口关系，而描述逻辑功能则比较烦琐。原理图输入对用户来讲很直观，尤其对表现层次结构、模块化结构更为方便。但它要求设计工具提供必要的元件库或逻辑宏单元。如果输入的是较为复杂的逻辑或元件库中不存在的模型，采用原理图输入往往很不方便。此外，原理图输入的设计可重用性、可移植性也差一些。

2. HDL 文本输入

硬件描述语言 (HDL) 是一种用文本形式来描述和设计电路的语言。设计者可利用 HDL 来描述自己的设计，然后利用 EDA 工具进行综合和仿真，最后变为某种目标文件，再用 ASIC 或 FPGA 具体实现。这种设计方法已被普遍采用。

硬件描述语言的发展至今不过几十年的历史，已成功应用于数字系统开发的各个阶段：设计、综合、仿真和验证等。到 20 世纪 80 年代，已出现了数十种硬件描述语言，但是，这些语言一般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，因此需要一种面向多领域、多层次并得到普遍认同的标准 HDL。进入 20 世纪 80 年代后期，硬件描述语言向着标准化、集成化的方向发展。最终，VHDL 和 Verilog 适应了这种趋势的要求，先后成为 IEEE 标准，在电子设计领域成为事实上的通用硬

件描述语言。

VHDL 和 Verilog 各有优点, 可用来进行算法级 (Algorithm Level)、寄存器传输级 (RTL)、门级 (Gate Level) 等各种层次的逻辑设计, 也可以进行仿真验证、时序分析等。由于 HDL 具有标准化特性, 易于将设计移植到不同厂家的芯片中去, 信号参数也容易改变和修改。此外, 采用 HDL 进行设计还具有工艺无关性, 这使得工程师在功能设计、逻辑验证阶段可以不必过多考虑门级及工艺实现的具体细节, 只需根据系统设计的要求, 施加不同的约束条件, 即可设计出实际电路。

PLD 器件的设计往往采用层次化的设计方法, 分模块、分层次地进行设计描述。描述器件总功能的模块放置在最上层, 称为顶层设计; 描述器件最基本功能的模块放置在最下层, 称为底层设计。顶层和底层之间的关系类似于软件中的主程序和子程序的关系。层次化设计的方法比较自由, 可以在任何层次使用原理图或硬件描述语言进行描述。一般做法是: 在顶层设计中, 使用图形法表达连接关系和芯片内部逻辑到引脚的接口; 在底层设计中, 使用硬件描述语言描述各个模块的逻辑功能。

1.3.2 综合

综合 (Synthesis) 是一个很重要的步骤, 综合指的是将较高级抽象层次的设计描述自动转化为较低层次描述的过程。综合有下面几种形式。

- 将算法表示、行为描述转换到寄存器传输级 (RTL), 即从行为描述到结构描述。
- 将 RTL 描述转换到逻辑门级 (包括触发器), 称为逻辑综合。
- 将逻辑门表示转换到版图表示, 或转换到 PLD 器件的配置网表表示; 根据版图信息能够进行 ASIC 生产, 有了配置网表, 可完成基于 PLD 器件的系统实现。

综合器就是能够自动实现上述转换的软件工具。或者说, 综合器是能够将原理图或 HDL 语言表达、描述的电路编译成由与或阵列、RAM、触发器、寄存器等逻辑单元组成的电路结构网表的工具。

硬件综合器和软件程序编译器是有本质区别的, 如图 1.7 所示为软件程序编译器和硬件综合器的比较, 软件程序编译器是将用 C 或汇编语言等编写的程序编译为 0、1 代码流, 而硬件综合器则是将用硬件描述语言编写的程序代码转化为具体的电路结构网表。

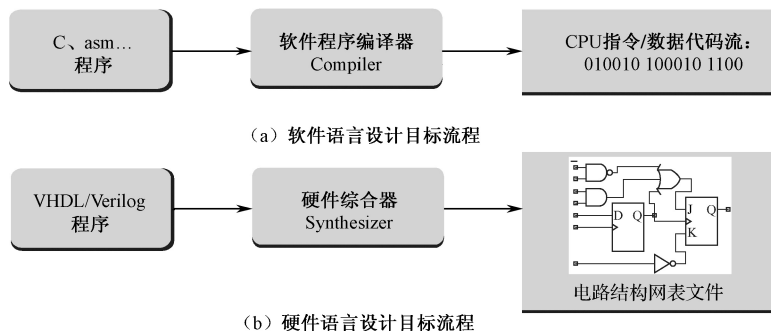


图 1.7 软件程序编译器和硬件综合器的比较

1.3.3 布局布线

布局布线 (Place & Route), 或者称为适配 (Fitting), 可理解为将综合生成的电路逻辑网表映射到具体的目标器件中实现, 并产生最终的可下载文件的过程。布局布线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, 把整个设计分为多个适合器件内部逻辑资源实现的逻辑小块, 并根据用户的设定在速度和面积之间做出选择或折中。布局是将已分割的逻辑小块放到器件内部逻辑资源的具

体位置,并使它们易于连线;布线则是利用器件的布线资源完成各功能块之间和反馈信号之间的连接。

布局布线完成后产生如下一些重要的文件。

(1) 芯片资源耗用情况报告。

(2) 面向其他 EDA 工具的输出文件,如 EDIF 文件等。

(3) 产生延时网表结构,以便于进行精确的时序仿真,因为已经提取出延时网表,所以仿真结果能比较精确地预测未来芯片的实际性能。如果仿真结果达不到设计要求,就需要修改源代码或选择不同速度的器件,直至满足设计要求。

(4) 器件编程文件:如用于 CPLD 编程的 JEDEC、POF 等格式的文件;用于 FPGA 配置的 SOF、JAM、BIT 等格式的文件。

由于布局布线与芯片的物理结构直接相关,因此一般选择芯片制造商提供的开发工具进行此项工作。

1.3.4 仿真

仿真(Simulation)也称为模拟,是对所设计电路的功能进行验证。用户可以在设计过程中对整个系统和各个模块进行仿真,即在计算机上用软件验证功能是否正确、各部分的时序配合是否准确。如果有问题可以随时进行修改,从而避免了逻辑错误。高级的仿真软件还可以对整个系统设计的性能进行估计。规模越大的设计,越需要进行仿真。

仿真包括功能仿真(Function Simulation)和时序仿真(Timing Simulation)。不考虑信号延时等因素的仿真称为功能仿真,又称前仿真;时序仿真又称后仿真,它是在选择具体器件并完成布局布线后进行的包含延时的仿真。由于不同器件的内部延时不一样,不同的布局、布线方案也给延时造成很大的影响,因此在设计实现后,对网络和逻辑块进行延时仿真、分析定时关系、估计设计性能是非常必要的。

1.3.5 编程配置

把适配后生成的编程文件装入 PLD 器件中的过程称为下载。通常将对基于 EEPROM 工艺的非易失结构 CPLD 器件的下载称为编程(Program),而将基于 SRAM 工艺结构的 FPGA 器件的下载称为配置(Configuration)。编程需要满足一定的条件,如编程电压、编程时序和编程算法等。有两种常用的编程方式:在系统编程(In-System Programmable, ISP)和使用专用的编程器编程,现在的 PLD 器件一般都支持在系统编程,因此在设计数字系统和做 PCB 时,应预留器件的下载接口。

1.4 常用的 EDA 工具软件

EDA 工具软件有两种分类方法:一种是按公司类别进行分类;另一种是按软件的功能进行分类。若按公司类别分,大体有两类:一类是专业 EDA 软件公司开发的工具,也称为第三方 EDA 软件工具,专业 EDA 公司比较著名的有 Cadence Design Systems、Mentor Graphics、Synopsys 和 Synplicity 四家,它们的软件工具被广泛地应用;另一类是 PLD 器件厂商为了销售其芯片而开发的 EDA 工具,较著名的有 Altera (Intel FPGA)、Xilinx、Lattice 等。前者独立于半导体器件厂商,其推出的 EDA 软件功能强,相互之间具有良好的兼容性,适合进行复杂和高效率的设计,但价格昂贵;后者能针对自己器件的工艺特点做出优化设计,提高资源利用率,降低功耗,改善性能,适合产品开发单位使用。

如果按功能分类,EDA 软件工具可分为如下几类。

1. 集成的 FPGA/CPLD 开发工具

集成的 FPGA/CPLD 开发工具是由 FPGA/CPLD 芯片生产厂家提供的, 这些工具可以完成从设计输入 (原理图或 HDL)、逻辑综合、模拟仿真到适配下载等全部工作。常用的集成 FPGA/CPLD 开发工具如表 1.1 所示, 这些开发工具多数将一些专业的第三方软件也集成在一起, 方便用户在设计过程中选择专业的第三方软件完成某些设计任务。

表 1.1 常用的集成 FPGA/CPLD 开发工具

软 件	说 明
	MAX+plus II 是 Altera 的集成开发软件, 使用广泛, 支持 Verilog、VHDL 和 AHDL, MAX+plus II 发展到 10.2 版本后, Altera 已不再推出新版本
	Quartus II 是 Altera 继 MAX+plus II 后的新一代开发工具, 适合大规模 FPGA 的开发。Quartus II 提供了更优化的综合和适配功能, 改善了对第三方仿真和时域分析工具的支持。Quartus II 还包含 DSP Builder、SOPC Builder 等开发工具, 支持系统级的开发, 支持 Nios II 嵌入式核、IP 核和用户定义逻辑等
	从 Quartus II 15.1 开始, Quartus II 改名为 Quartus Prime。2016 年 5 月 Intel (2015 年 Altera 被 Intel 收购) 发布了 Quartus Prime 16.0, 分为 Pro、Standard、Lite 三个版本。目前, Quartus Prime 已发布的最新版本是 17.1。Quartus Prime 软件中集成了新的 Spectra-Q 综合工具, 支持数百万 LE 单元的 FPGA 器件的综合; 集成了新的前端语言解析器, 扩展了对 VHDL-2008 和 SystemVerilog-2005 的支持, 增强了 RTL 设计
	ISE 是 Xilinx 公司 FPGA/CPLD 的集成开发软件, 它提供给用户从设计输入到综合、布线、仿真、下载的全套解决方案, 并很方便地同其他 EDA 工具接口。其中, 原理图输入用的是第三方软件 ECS, HDL 综合可以使用 Xilinx 公司开发的 XST、Synopsys 的 FPGA Express 和 Synplicity 的 Synplify/Synplify Pro, 测试输入是图形化的 HDL Bench, 状态图输入用的是 StateCAD, 前、后仿真则可以使用 ModelSim XE 或 ModelSim SE
	Vivado 设计套件, 是 FPGA 厂商 Xilinx 公司 2012 年发布的集成设计环境。包括高度集成的设计环境和新一代从系统到 IC 级的工具, 这些均建立在共享的可扩展数据模型和通用调试环境基础上。这也是一个基于 AMBA AXI4 互连规范、IP-XACT IP 封装元数据、工具命令语言 (TCL)、Synopsys 系统约束 (SDC) 及其他有助于根据客户需求量身定制设计流程并符合业界标准的开放式环境, 支持多达一亿个等效 ASIC 门的设计
	ispLEVER Classic 是 Lattice 公司的 FPGA 设计环境, 支持 FPGA 器件的整个设计过程, 从概念设计到器件 JEDEC 或位流编程文件输出。当前版本是 ispLEVER Classic 2.0, 于 2015 年 6 月 16 日发布, 支持 Windows 7、Windows Vista 和 Windows XP 等操作系统
	Diamond 软件是 Lattice 公司的开发工具, 支持 FPGA 从设计输入到位流下载的整个流程。支持 Windows 7、Windows 8 等操作系统

2. 设计输入工具

输入工具主要是帮助用户完成原理图和 HDL 文本的编辑及输入工作。好的输入工具能够支持多种输入方式，包括原理图、HDL 文本、波形图、状态机、真值表等。例如，HDL Designer Series 是 Mentor 公司的设计输入工具，包含在 FPGA Advantage 软件中，可以接受 HDL 文本、原理图、状态图、表格等多种设计输入形式，并将其转化为 HDL 文本表达方式，功能很强。输入工具可帮助用户提高输入效率，多数人习惯使用集成开发软件或综合/仿真工具中自带的原理图和文本编辑器，也可以直接使用普通文本编辑器，如 Notepad++ 等。

3. 逻辑综合器 (Synthesizer)

逻辑综合是将设计者在 EDA 平台上编辑输入的 HDL 文本、原理图或状态图描述，依据给定的硬件结构和约束控制条件进行编译、优化及转换，最终获得门级电路甚至底层的电路描述网表文件的过程。




逻辑综合工具能够自动完成上述过程，产生优化的电路结构网表，输出 .edf 文件，给 FPGA/CPLD 厂家的软件进行适配和布局布线。专业的逻辑综合软件通常比 FPGA/CPLD 厂家的集成开发软件中自带的逻辑综合功能更强，能得到更优化的结果。

最著名的用于 FPGA/CPLD 设计的 HDL 综合工具有如下 3 个：

- Synopsys 公司的 FPGA Express、FPGA Compiler 和 FPGA Compiler II；
- Synplicity (Synplicity 已被 Synopsys 收购) 的 Synplify Pro/Synplify；
- Mentor 的 Leonardo Spectrum。

表 1.2 所示为常用的 HDL 综合工具。

表 1.2 常用的 HDL 综合工具

软 件	说 明
	Synplify Pro/Synplify 是 Synplicity (已被 Synopsys 收购) 的 VHDL/Verilog 综合软件，使用广泛。Synplify Pro 除具有原理图生成器、延时分析器外，还带了一个 FSM Compiler (有限状态机编译器)，能从 HDL 设计文本中提出存在的 FSM 设计模块，并用状态图的方式显示出来
	FPGA Compiler II 是 Synopsys 公司的 VHDL/Verilog 综合软件。Synopsys 是最早推出 HDL 综合器的公司，它改变了早先 HDL 只能用于电路的模拟仿真的状况。Synopsys 的综合器包括 FPGA Express、FPGA Compiler，目前其最新的综合软件为 FPGA Compiler II
	Leonardo Spectrum 是 Mentor 的子公司 Exemplar Logic 出品的 VHDL/Verilog 综合软件，并作为 FPGA Advantage 软件的一个组成部分。Leonardo Spectrum 可同时用于 FPGA/CPLD 和 ASIC 设计，性能稳定

4. 仿真器




仿真工具提供了对设计进行模拟仿真的手段，包括布线以前的功能仿真（前仿真）和布线以后包含延时的时序仿真（后仿真）。在一些复杂的设计中，仿真比设计本身还要艰巨，因此有人认为仿真是 EDA 的精髓所在，仿真器的仿真速度、仿真的准确性、易用性等成为衡量仿真器性能的重要指标。

按对设计语言的不同处理方式，仿真器分为两类：编译型仿真器和解释型仿真器。编译型仿真器的仿真速度快，但需要预处理，因此不能即时修改；解释型仿真器的仿真速度要慢一些，但可以随时

修改仿真环境和仿真条件。按处理的 HDL 类型, 仿真器可分为 Verilog 仿真器、VHDL 仿真器和混合仿真器, 混合仿真器能够同时处理 Verilog 和 VHDL。

常用的 HDL 仿真软件如表 1.3 所示。

表 1.3 常用的 HDL 仿真软件

软 件	说 明
 ModelSim	ModelSim 是 Mentor 的子公司 Model Technology 的一个出色的 VHDL/Verilog 混合仿真软件, 它属于编译型仿真器, 仿真速度快, 功能强
 NC-Verilog/NC-VHDL/NC-Sim Verilog-XL	这几个软件都是 Cadence 公司的 VHDL/Verilog 仿真工具, 其中 NC-Verilog 的前身是著名的 Verilog 仿真软件 Verilog-XL, 用于对 Verilog 程序进行仿真; NC-VHDL 用于 VHDL 仿真; 而 NC-Sim 则能够对 VHDL/Verilog 进行混合仿真
 VCS/Scirocco	VCS 是 Synopsys 公司的 Verilog 仿真软件, Scirocco 是 Synopsys 的 VHDL 仿真软件
Active HDL	Active HDL 是 Aldec 的 VHDL/Verilog 仿真软件, 简单易用

ModelSim 能够提供很好的 Verilog 和 VHDL 混合仿真; NC-Verilog 和 VCS 是基于编译技术的仿真软件, 能够胜任行为级、RTL 和门级各种层次的仿真, 速度快; 而 Verilog-XL 是基于解释的仿真工具, 速度要慢一些。


5. 芯片版图设计软件

提供 IC 版图设计工具的著名公司有 Cadence、Mentor、Synopsys 等, Synopsys 的优势在于其逻辑综合工具, 而 Mentor 和 Cadence 则能够在设计的各个层次提供全套的开发工具。在晶体管级或基本门级提供图形输入工具的有 Cadence 的 Composer、Viewlogic 公司的 Viewdraw 等。专用于 IC 的综合工具有 Synopsys 的 Design Compiler、Behavioral Compiler, Synplicity 的 Synplify ASIC, Cadence 的 Synergy 等。IC 仿真工具的关键在于晶体管物理模型的建立和实际工艺中晶体管物理特性相符的模型必然得到和实际电路更符合的工作波形。随着 IC 集成度的日益提高、线宽的日趋缩小, 晶体管的模型也日趋复杂。任何电路仿真都基于一定的厂家库, 在这些库文件中制造厂家为设计者提供了相应的工艺参数。SPICE 是著名的模拟电路仿真工具, SPICE 最早产生于 Berkley 大学, 经历数年的发展, 随晶体管线宽的不断缩小, SPICE 也引入了更多的参数和更复杂的晶体管模型, 使其在亚微米和深亚微米工艺的今天依旧是模拟电路仿真的重要工具之一。此外, 还有一些其他 IC 版图工具, 如自动布局布线 (Auto Plane & Route) 工具、版图输入工具、物理验证 (Physical Validate) 和参数提取 (LVS) 工具等。半导体集成技术还在不断地发展, 相应的 IC 设计工具也不断地更新换代, 以提供对 IC 设计的全方位支持, 应该说没有 EDA 工具, 就没有 IC。

6. 其他 EDA 专用工具

除上面介绍的 EDA 软件外, 一些公司还推出了一些开发套件和专用的开发工具, 比如 Quartus Prime 推出的 Platform Designer 就是一种基于 PBD (Platform Based Design) 设计理念的开发工具, 它是一种基于 IP 的面向 SoC 的设计环境, 可以在更短的时间内设计出满足需要的电路。专用的 EDA 开发套件和开发工具如表 1.4 所示。

表 1.4 专用的 EDA 开发套件和开发工具

软 件	说 明
	Mentor 公司的 VHDL/Verilog 完整开发系统, 可以完成除适配和编程外的所有工作, 包括三套软件: HDL Designer Series (输入及项目管理), Leonardo Spectrum (逻辑综合) 和 ModelSim (模拟仿真)
DSP Builder	Altera 的 DSP 开发工具, 设计者可以在 MATLAB 和 Simulink 软件中进行高级抽象层的 DSP 算法设计, 然后自动将算法设计转化为 HDL 文件, 实现了从常用 DSP 开发工具 (MATLAB) 到 EDA 工具 (Quartus II) 的无缝连接。DSP Builder 还能够生成 SOPC Builder Ready DSP 模块, 采用 SOPC Builder 可将其集成到一个完整的 SOPC 系统设计中
SOPC Builder Qsys Platform Designer	自从 Quartus II 10 之后, SOPC Builder 就被 Qsys 代替了, Qsys 则是 SOPC Builder 的升级版, 用于系统级的 IP 集成, 能将不同的 IP 模块及 Nios II 核方便快捷地整合成一个系统, 提高 FPGA 设计的效率; 从 Quartus Prime 17.1 版开始, Qsys 更名为 Platform Designer, 内容与名字更为统一
System Generator	Xilinx 的 DSP 开发工具, 实现 ISE 与 MATLAB 的接口, 能有效地完成数字信号处理的仿真和最终 FPGA 实现

1.5 EDA 技术的发展趋势

1. 高性能的 EDA 工具将得到进一步发展

随着市场需求的增长, 以及集成工艺水平及计算机自动设计技术的不断提高, 单片系统或系统集成芯片成为 IC 设计的主流, 这一发展趋势表现在以下几个方面。

(1) 随着超大规模集成电路技术水平的不断提高, 超深亚微米 (VDSM) 工艺 (如 14nm) 已经走向成熟, 在一个芯片上完成系统级的集成已成为现实。

(2) 由于工艺线宽不断减小, 在半导体材料上的许多寄生效应已经不能简单地被忽略, 这就对 EDA 工具提出了更高的要求。同时, 也使得 IC 生产线的投资更为巨大, 可编程逻辑器件开始进入传统的 ASIC 市场。

(3) 市场对电子产品提出更高的要求, 如必须降低电子系统的成本、减小系统的体积等, 从而对系统的集成度不断提出更高的要求。同时, 设计效率也成为产品能否成功的关键因素, 使得 EDA 工具和 IP 核应用更为广泛。

(4) 高性能的 EDA 工具将得到长足的发展, 其自动化和智能化程度将不断提高, 从而为嵌入式系统设计提供功能强大的开发环境。此外, 计算机硬件平台性能的大幅度提高, 也为复杂的 SoC 设计提供了物质基础。

由于现在的硬件描述语言只提供行为级或功能级的描述, 尚无法完成系统级的抽象描述, 因此人们正尝试开发一种新的系统级设计语言来完成这一工作, 现在已经开发出更趋于电路行为级设计的硬件描述语言, 如 SystemC、System Verilog 等。SystemC 由 Synopsys 公司和 CoWare 公司合作开发, 目前已有一些 EDA 公司成立“开放式 SystemC 联盟”, 支持 SystemC 的开发。还出现了一些系统级混合仿真工具, 可以在同一个开发平台上完成高级语言 (如 C/C++ 等) 与标准硬件描述语言 (Verilog、VHDL) 的混合仿真。

2. EDA 技术将促使 ASIC 和 FPGA 逐步走向融合

此外,随着系统开发对 EDA 技术的目标器件各种性能指标要求的提高,ASIC 和 FPGA 将更程度地相互融合。这是因为虽然标准逻辑 ASIC 芯片尺寸小、功能强、耗电小,但设计复杂,并且有批量生产要求;可编程逻辑器件的开发费用低廉,能现场编程,但体积大、功耗大。因此 FPGA 和 ASIC 正在走到一起,两者之间正在诞生一种“杂交”产品,互相融合,取长补短,以满足成本和上市速度的要求,例如,将可编程逻辑器件嵌入标准单元。

3. EDA 技术的应用领域将更为广泛

从目前的 EDA 技术来看,其特点是使用普及、应用面广、工具多样。在 ASIC 和 PLD 器件方面,正在向超高速、高密度、低功耗、低电压方向发展。EDA 技术水平不断进步,设计工具不断趋于完善。

习 题 1

- 1.1 现代 EDA 技术的特点有哪些?
- 1.2 什么是 Top-down 设计方式?
- 1.3 数字系统的实现方式有哪些?各有什么优缺点?
- 1.4 什么是 IP 复用技术?IP 核对 EDA 技术的应用和发展有什么意义?
- 1.5 用硬件描述语言设计数字电路的优势是什么?
- 1.6 结合自己的使用情况谈谈对 EDA 工具的认识。
- 1.7 基于 FPGA/CPLD 的数字系统设计流程包括哪些步骤?
- 1.8 什么是综合?常用的综合工具有哪些?
- 1.9 功能仿真与时序仿真有什么区别?
- 1.10 FPGA 与 ASIC 在概念上有什么区别?