

# 集成电路设计

维基百科，自由的百科全书

**集成电路设计**（英語：**Integrated circuit design, IC design**），根据当前集成电路的集成规模，亦可称之为**超大规模集成电路设计**（**VLSI design**）<sup>[1]</sup>，是指以集成电路、超大规模集成电路为目标的设计流程。

<b>目录</b>
<b><u>概述</u></b>
<b><u>设计的抽象级别</u></b>
<b><u>硬件实现</u></b>
<u>全定制设计</u>
<u>半定制设计</u>
<u>可编程逻辑器件</u>
<u>专用集成电路</u>
<b><u>设计流程</u></b>
<u>模拟集成电路</u>
<u>数字集成电路</u>
<u>系统定义</u>
<u>寄存器传输级设计</u>
<u>设计验证</u>
<u>逻辑综合</u>
<u>等效性检查</u>
<u>时序分析</u>
<u>物理设计</u>
<u>后续：具体的工艺制造</u>
<b><u>可测试性设计</u></b>
<b><u>设计的重用</u></b>
<b><u>设计的辅助和自动化</u></b>
<b><u>参考文献</u></b>
<b><u>参见</u></b>

# 概述

---

集成电路设计涉及对电子器件（例如晶体管、电阻器、电容器等）、器件间互连线模型的建立。所有的器件和互连线都需安置在一块半导体衬底材料之上，这些元件通过半导体器件制造工艺（例如光刻等）安置在单一的硅衬底上，从而形成电路。<sup>[2]:7-10</sup>目前最常使用的衬底材料是硅。设计人员会使用技术手段将硅衬底上各个器件之间相互电隔离，以控制整个芯片上各个器件之间的导电性能。PN结、金属氧化物半导体场效应晶体管等组成了集成电路器件的基础结构，而由后者构成的互补式金属氧化物半导体则凭借其低静态功耗、高集成度的优点成为数字集成电路中逻辑门的基础构造。<sup>[2]:197-198</sup>设计人员需要考虑晶体管、互连线的能量耗散，这一点与以往由分立电子器件开始构建电路不同，这是因为集成电路的所有器件都集成在一块硅片上。金属互连线的电迁移以及静电放电对于微芯片上的器件通常有害，因此也是集成电路设计需要关注的课题。

随着集成电路的规模不断增大，其集成度已经达到深亚微米级（特征尺寸在130纳米以下），单个芯片集成的晶体管已经接近十亿个。<sup>[3]:48-49</sup>由于其极为复杂，集成电路设计相较于简单电路设计常常需要计算机辅助的设计方法学和技术手段。集成电路设计的研究范围涵盖了数字集成电路中数字逻辑的优化、网表实现，寄存器传输级硬件描述语言代码的书写，逻辑功能的验证、仿真和时序分析，电路在硬件中连线的分布，模拟集成电路中运算放大器、电子滤波器等器件在芯片中的安置和混合信号的处理。相关的研究还包括硬件设计的电子设计自动化（EDA）、计算机辅助设计（CAD）方法学等，是电机工程学和计算机工程的一个子集。

对于数字集成电路来说，设计人员现在更多的是站在高级抽象层面，即寄存器传输级甚至更高的系统级（有人也称之为行为级），使用硬件描述语言或高级建模语言来描述电路的逻辑、时序功能，而逻辑综合可以自动将寄存器传输级的硬件描述语言转换为逻辑门级的网表。对于简单的电路，设计人员也可以用硬件描述语言直接描述逻辑门和触发器之间的连接情况。网表经过进一步的功能验证、布局、布线，可以产生用于工业制造的GDSII文件，工厂根据该文件就可以在晶圆上制造电路。<sup>[4]:388-389</sup>模拟集成电路设计涉及了更加复杂的信号环境，对工程师的经验有更高的要求，并且其设计的自动化程度远不及数字集成电路。<sup>[5]:3</sup>

逐步完成功能设计之后，设计规则会指明哪些设计符合制造要求，而哪些设计不符合，而这个规则本身也十分复杂。集成电路设计流程需要符合数百条这样的规则。在一定的设计约束下，集成电路物理版图的布局、布线对于获得理想速度、信号完整性、减少芯片面积来说至关重要。半导体器件制造的不可预测性使得集成电路设计的难度进一步提高。在集成电路设计领域，由于市场竞争的压力，电子设计自动化等相关计算机辅助设计工具得到了广泛的应用，工程师可以在计算机软件的辅助下进行寄存器传输级设计、功能验证、静态时序分析、物理设计等流程。<sup>[6]:2-3</sup>

## 设计的抽象级别

---

集成电路设计通常是以“模块”作为设计的单位的。例如，对于多位全加器来说，其次级模块是一位的加法器，而加法器又是由下一级的与门、非门模块构成，与、非门最终可以分解为更低抽象级的CMOS器件。

从抽象级别来说，数字集成电路设计可以是自顶向下的，即先定义了系统最高逻辑层次的功能模块，根据顶层模块的需求来定义子模块，然后逐层继续分解；设计也可以是自底向上的，即先分别设计最具体的各个模块，然后如同搭积木一般用这些最底层模块来实现上层模块，最终达到最高层次。<sup>[7]:257</sup>在许多设计中，自顶向下、自底向上的设计方法学是混合使用的，系统级设计人员对整体体系结构进行规划，并进行子模块的划分，而底层的电路设计人员逐层向上设计、优化单独的模块。最后，两个方向的设计人员在中间某一抽象层次会合，完成整个设计。<sup>[8]:9</sup>

## 硬件实现

---

对于不同的设计要求，工程师可以选择使用半定制设计途径，例如采用可编程逻辑器件（现场可编程逻辑门阵列等）或基于标准单元库的特殊应用集成电路来实现硬件电路；也可以使用全定制设计，控制晶体管版图到系统结构的全部细节。<sup>[9]:2-6</sup>

## 全定制设计

这种设计方式要求设计人员利用版图编辑器来完成版图设计、参数提取、单元表征，然后利用这些自己设计的单元来完成电路的构建。通常，全定制设计是为了最大化优化电路性能。如果标准单元库中缺少某种所需的单元，也需要采取全定制设计的方法完成所需的单元设计。不过，这种设计方式通常需要较长的时间。<sup>[10]:3</sup>

## 半定制设计

与全定制设计相对的设计方式为半定制设计。简而言之，半定制集成电路设计是基于预先设计好的某些逻辑单元。例如，设计人员可以在标准元件库（通常可以从第三方购买）的基础上设计专用集成电路，从中选取所需的逻辑单元（例如各种基本逻辑门、触发器等）来搭建所需的电路。<sup>[10]:4</sup>他们也可以使用可编程逻辑器件来完成设计，这类器件的几乎所有物理结构都已经固定在芯片之中，只剩下某些连线可以由用户编程决定其连接方式。<sup>[10]:9</sup>与这些预先设计好的逻辑单元有关的性能参数通常也由其供应商提供，以方便设计人员进行时序、功耗分析。

## 可编程逻辑器件

可编程逻辑器件通常由半导体厂家提供商品芯片，这些芯片可以通过JTAG等方式和计算机连接，因此设计人员可以用电子设计自动化工具来完成设计，然后将利用设计代码来对逻辑芯片编程。可编程逻辑阵列芯片在出厂前就提前定义了逻辑门构成的阵列，而逻辑门之间的连接线路则可以通过编程来控制连接与断开。<sup>[11]:197</sup>随着技术的发展，对连接线的编程可以通过EPROM（利用较高压电编程、紫外线照射擦除）、EEPROM（利用电信号来多次编程和擦除）、SRAM、闪存等方式实现。现场可编程逻辑门阵列是一种特殊的可编程逻辑器件，它的物理基础是可配置逻辑单元，由查找表、可编程多路选择器、寄存器等结构组成。查找表可以用来实现逻辑函数，如三个输入端的查找表可以实现所有三变量的逻辑函数。<sup>[11]:199</sup>



在半定制的现场可编程逻辑门阵列（FPGA）上实现设计的优点是开发周期短、成本低。

## 专用集成电路

专用集成电路只能在整个集成电路设计完成之后才能开始制造，而且需要专业的半导体工厂的参与。专用集成电路可以是基于标准单元库，也可以是全定制设计。在后一种途径中，设计人员对于晶圆上元件的位置和连接有更多的控制权，而不像可编程逻辑器件途径，只能选择使用其中部分硬件资源，从而造成部分资源被浪费。专用集成电路的面积、功耗、时序特性通常可以得到更好的优化。然而，专用集成电路的设计会更加复杂，并且需要专门的工艺制造部门（或者外包给晶圆代工厂）才能将GDSII文件制造成电路。一旦专用集成电路芯片制造完成，就不能像可编程逻辑器件那样对电路的逻辑功能进行重新配置。对于单个产品，在专用集成电路上实现集成电路的经济、时间成本都比可编程逻辑器件高，因此在早期的设计与调试过程中，常用可编程逻辑器件，尤其是现场可编程逻辑门阵列；如果所设计的集成电路将要在后期大量投产，那么批量生产专用集成电路将会更经济。<sup>[9]:103-106</sup>

# 设计流程

集成电路设计可以大致分为数字集成电路设计和模拟集成电路设计两大类。不过，实际的集成电路还有可能是混合訊號積體電路，因此不少电路的设计同时用到这两种流程。

## 模拟集成电路

集成电路设计的另一个大分支是模拟集成电路设计，这一分支通常关注电源集成电路、射频集成电路等。<sup>[3]:7</sup>由于现实世界的信号是模拟的，所以，在电子产品中，模-数、数-模相互转换的集成电路也有着广泛的应用。模拟集成电路包括运算放大器、线性整流器、锁相环、振荡电路、有源滤波器等。相较数字集成电路设计，模拟集成电路设计与半导体器件的物理性质有着更大的关联，例如其增益、电路匹配、功率耗散以及阻抗等等。模拟信号的放大和滤波要求电路对信号具备一定的保真度，因此模拟集成电路比数字集成电路使用了更多的大面积器件，集成度亦相对较低。

在微处理器和计算机辅助设计方法出现前，模拟集成电路完全采用人工设计的方法。由于人处理复杂问题的能力有限，因此当时的模拟集成电路通常是较为基本的电路，运算放大器集成电路就是一个典型的例子。在当时的情况下，这样的集成电路可能会涉及十几个晶体管以及它们之间的互连线。为了使模拟集成电路的设计能达到工业生产的级别，工程师需要采取多次迭代的方法以测试、排除故障。重复利用已经设计、验证的设计，可以进一步构成更加复杂的集成电路。1970年代之后，计算机的价格逐渐下降，越来越多的工程师可以利用这种现代的工具来辅助设计，例如，他们使用编好的计算机程序进行仿真，便可获得比之前人工计算、设计更高的精确度。SPICE是第一款针对模拟集成电路仿真的软件（事实上，数字集成电路中标准单元本身的设计，也需要用到SPICE来进行参数测试），其字面意思是“以集成电路为重点的仿真程序（英語：Simulation Program with Integrated Circuit Emphasis）”。<sup>[12]</sup>基于计算机辅助设计的电路仿真工具能够适应更加复杂的现代集成电路，特别是专用集成电路。使用计算机进行仿真，还可以使项目设计中的一些错误在硬件制造之前就被发现，从而减少因为反复测试、排除故障造成的大量成本。此外，计算机往往能够完成一些极端复杂、繁琐，人类无法胜任的任务，使得诸如蒙地卡羅方法等成为可能。实际硬件电路会遇到的与理想情况不一致的偏差，例如温度偏差、器件中半导体掺杂浓度偏差，计算机仿真工具同样可以进行模拟和处理。总之，计算机化的电路设计、仿真能够使电路设计性能更佳，而且其可制造性可以得到更大的保障。尽管如此，相对数字集成电路，模拟集成电路的设计对工程师的经验、权衡矛盾等方面的能力要求更严格。

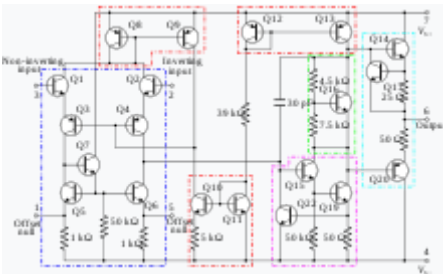
## 数字集成电路

粗略地说，数字集成电路可以分为以下基本步骤：系统定义、寄存器传输级设计、物理设计。而根据逻辑的抽象级别，设计又分为系统行为级、寄存器传输级、逻辑门级。<sup>[1]:29-32</sup><sup>[13]:13</sup>设计人员需要合理地书写功能代码、设置综合工具、验证逻辑时序性能、规划物理设计策略等等。在设计过程中的特定时间点，还需要多次进行逻辑功能、时序约束、设计规则方面的检查、调试，以确保设计的最终成果合乎最初的设计收敛目标。<sup>[14]:3-4</sup>

## 系统定义



针对特殊应用设计的专用集成电路（ASIC）的优点是面积、功耗、时序可以得到最大程度地优化。



运算放大器本身也常常是具有一定规模集成度的电路。

系统定义是进行集成电路设计的最初规划，在此阶段设计人员需要考虑系统的宏观功能。设计人员可能会使用一些高抽象级建模语言和工具来完成硬件的描述，例如C语言、C++、SystemC、SystemVerilog等事务级建模语言，以及Simulink和MATLAB等工具对信号进行建模。尽管目前的主流是以寄存器传输级设计为中心，但已有一些直接从系统级描述向低抽象级描述（如逻辑门级结构描述）转化的高级综合（或称行为级综合）、高级验证工具正处于发展阶段。<sup>[15]</sup>

系统定义阶段，设计人员还对芯片预期的工艺、功耗、时脉频率、工作温度等性能指标进行规划。

## 寄存器传输级设计

目前的集成电路设计常常在寄存器传输级上进行，利用硬件描述语言来描述数字集成电路的信号储存以及信号在寄存器、存储器、组合逻辑装置和总线等逻辑单元之间传输的情况。在设计寄存器传输级代码时，设计人员会将系统定义转换为寄存器传输级的描述。设计人员在这一抽象层次最常使用的两种硬件描述语言是Verilog、VHDL，二者分别于1995年和1987年由电气电子工程师学会（IEEE）标准化。<sup>[11]:110</sup>正由于有着硬件描述语言，设计人员可以把更多的精力放在功能的实现上，这比以往直接设计逻辑门级连线的方法学（使用硬件描述语言仍然可以直接设计门级网表，但是少有人如此工作）具有更高的效率。

## 设计验证

设计人员完成寄存器传输级设计之后，会利用测试平台、断言等方式来进行功能验证，检验项目设计是否与之前的功能定义相符，如果有误，则需要检测之前设计文件中存在的漏洞。<sup>[16]</sup>现代超大规模集成电路的整个设计过程中，验证所需的时间和精力越来越多，甚至都超过了寄存器传输级设计本身，人们设置些专门针对验证开发了新的工具和语言。

例如，要实现简单的加法器或者更加复杂的算术逻辑单元，或利用触发器实现有限状态机，设计人员可能会编写不同规模的硬件描述语言代码。功能验证是项复杂的任务，验证人员需要为待测设计建立一个虚拟的外部环境，为待测设计提供输入信号（这种人为添加的信号常用“激励”这个术语来表示），然后观察待测设计输出端口的功能是否合乎设计规范。

当所设计的电路并非简单的几个输入端口、输出端口时，由于验证需要尽可能地考虑到所有的输入情况，因此对于激励信号的定义会变得更加复杂。有时工程师会使用某些脚本语言（如Perl、Tcl）来编写验证程序，借助计算机程序的高速处理来实现更大的测试覆盖率。现代的硬件验证语言可以提供一些专门针对验证的特性，例如带有约束的随机化变量、覆盖等等。作为硬件设计、验证统一语言，SystemVerilog是以Verilog为基础发展而来的，因此它同时具备了设计的特性和测试平台的特性，并引入了面向对象程序设计的思想，因此测试平台的编写更加接近软件测试。<sup>[17]</sup>诸如通用验证方法学的标准化验证平台开发框架也得到了主流电子设计自动化软件厂商的支持。针对高级综合，关于高级验证的电子设计自动化工具也处于研究中。

## 逻辑综合

工程师设计的硬件描述语言代码一般是寄存器传输级的，在进行物理设计之前，需要使用逻辑综合工具将寄存器传输级代码转换到针对特定工艺的逻辑门级网表，并完成逻辑化简。

和人工进行逻辑优化需要借助卡诺图等类似，电子设计自动化工具来完成逻辑综合也需要特定的算法（如奎因一麦克拉斯基算法等）来化简设计人员定义的逻辑函数。输入到自动综合工具中的文件包括寄存器传输级硬件描述语言代码、工艺库（可以由第三方晶圆代工服务机构提供）、设计约束文件三大类，这些文件在不同的电子设计自动化工具套件系统中的格式可能不尽相同。逻辑综合工具会产生一个优化后的门级网表，但是这个网表仍然是基于硬件描述语言的，这个网表在半导体芯片中的走线将在物理设计来完成。<sup>[8]:201-204</sup>



选择不同器件（如专用集成电路或者现场可编程门阵列等）对应的工艺库来进行逻辑综合，或者在综合时设置了不同的约束策略，将产生不同的综合结果。寄存器传输级代码对于设计项目的逻辑计划、语言结构风格等因素会影响综合后网表的效率。

目前大多数成熟的综合工具大多数是基于寄存器传输级描述的，而基于系统级描述的高级综合工具还处在发展阶段。

## 等效性检查

为了比较门级网表和寄存器传输级的等效性，可以通过生成诸如不二可满足性、二元决策图等途径来完成形式等效性检查（形式验证）。实际上，等效性检查还可以检查两个寄存器传输级设计之间，或者两个门级网表之间的逻辑等效性。

## 时序分析

现代集成电路的时钟频率已经到达了兆赫兹级别，而大量模块内、模块之间的时序关系极其复杂，<sup>[13]:7-8</sup>因此，除了需要验证电路的逻辑功能，还需要进行时序分析，即对信号在传输路径上的延迟进行检查，判断其是否符合时序收敛要求。<sup>[14]:7</sup>时序分析所需的逻辑门标准延迟格式信息可以由标准单元库（或从用户自己设计的单元中提取的时序信息）提供。随着电路特征尺寸不断减小，互连线延迟在实际的总延时中所占的比例愈加显著，因此在物理设计完成之后，把互连线的延迟纳入考虑，才能够精准地进行时序分析。

## 物理设计

逻辑综合完成之后，通过引入器件制造公司提供的工艺信息，前面完成的设计将进入布图规划、布局、布线阶段，工程人员需要根据延迟、功耗、面积等方面的约束信息，合理设置物理设计工具的参数，不断调试，以获取最佳的配置，从而决定元件在晶圆上的物理位置。如果是全定制设计，工程师还需要精心绘制单元的集成电路版图，调整晶体管尺寸，从而降低功耗、延时。<sup>[14]:8-9</sup>

随着现代集成电路的特征尺寸不断下降，超大规模集成电路已经进入深亚微米级阶段，互连线延迟对电路性能的影响已经达到甚至超过逻辑门延迟的影响。这时，需要考虑的因素包括线网的电容效应和线网电感效应，芯片内部电源线上大电流在线网电阻上造成的电压降也会影响集成电路的稳定性。<sup>[13]:6-7</sup>为了解决这些问题，同时缓解时钟偏移、时钟树寄生参数的负面影响，合理的布局布线和逻辑设计、功能验证等过程同等重要。<sup>[3]:49</sup>随着移动设备的发展，低功耗设计在集成电路设计中的地位愈加显著。在物理设计阶段，设计可以转化成几何图形的表示方法，工业界有若干标准化的文件格式（如GDSII）予以规范。

值得注意的是，电路实现的功能在之前的寄存器传输级设计中就已经确定。在物理设计阶段，工程师不仅不能够让之前设计好的逻辑、时序功能在该阶段的设计中被损坏，还要进一步优化芯片按照正确运行时的延迟时间、功耗、面积等方面的性能。在物理设计产生了初步版图文件之后，工程师需要再次对集成电路进行功能、时序、设计规则、信号完整性等方面的验证，以确保物理设计产生正确的硬件版图文件。<sup>[14]</sup>

## 后续：具体的工艺制造

半导体制造工厂根据物理设计最后完成、已经通过各项检查后生成的标准化GDSII文件，即可制造出实际的物理电路。

这个步骤不再属于集成电路设计和计算机工程的范畴，而是直接进入半导体制造工艺领域，关注的重心亦转向具体的材料、器件制作，例如光刻、刻蚀、物理气相沉积、化学气相沉积等。

传统的集成电路公司能够同时完成集成电路设计和集成电路制造。由于集成电路制造所需的设备、原料耗资巨大，因此一般的公司根本无力承受。一旦发生工艺节点的改变（如从65纳米工艺进步到45纳米工艺），公司可能需要花费相当高的成本来更换现有工艺设备，这给许多公司带来了相当沉重的经济负担。现在，有些公司逐渐放弃既设计、又制造的模式，业务范围缩小至设计、验证本身，而将具体的半导体工艺流程，委托给专门进行集成电路制造的工厂。上述无制造工艺（fabless），只进行设计、验证公司被称为無廠半導體公司，典型的例子包括高通（Qualcomm）、超微半导体（AMD）、英伟达（NVIDIA）等；而专门负责制造的公司则被称为晶圆代工厂，典型的例子包括台积电（TSMC）、格羅方德（GlobalFoundries，前身为AMD的直属工艺厂）等。某些公司在从事设计的同时，还保留了自己的工艺厂，这样的公司包括英特尔、三星電子等。还有一类特殊的无厂半导体公司，它们把设计项目以IP核的形式封装起来，作为商品销售给其他无厂半导体公司，典型的例子包括ARM公司。

## 可测试性设计

---

随着超大规模集成电路的复杂程度不断提高，电路制造后的测试所需的时间和经济成本也不断增加。以往，人们将绝大多数精力放在设计本身，而并不考虑之后的测试，因为那时的测试相对今天更为简单。近年来，测试本身也逐渐成为一个庞大的课题。

比如，从电路外部控制某些内部信号使得它们呈现特定的逻辑值比较容易，而某些内部信号由于依赖大量其它内部信号，从外部很难直接改变它们的数值。此外，内部信号的改变很多时候不能在主输出端观测（有时主输出端的信号输出看似正确，其实内部状态是错误的，仅观测主输出端的输出不足以判断电路是否正常工作）。以上两类问题，即可控制性和可观测性，是可测试性的两大组成部分。

人们逐渐发现，电路在设计时向电路添加一些特殊的结构（例如扫描链和内建自测试），能够大大方便之后的电路测试。这样的设计被即为可测试性设计，它们使电路更加复杂，但是却能凭借更简捷的测试降低整个项目的成本。<sup>[18]</sup>

## 设计的重用

---

随着超大规模集成电路的集成難度不断提高，同时市场竞争压力不断增加，集成电路设计逐渐引入了可重用设计方法学<sup>[19]</sup>。

可重用设计方法学的主要意义在于提供IP核（知识产权核）的供应商可以将一些已预先完成之设计以商品的形式提供给设计方，后者可以将IP核作为一个完整的模块在自己的设计项目中使用。由此，在实现类似功能时，各个公司就不需反复设计类似模块。这样做虽会提高商业成本，但亦显著降低了设计的复杂程度，从而缩短设计大型电路所需的時間周期並提高市场竞争力。

IP核供应商提供的产品可能是已验证的硬件描述语言代码，为了保护供应商的知识产权，这些代码大多是加密的。IP核本身也是作为集成电路进行设计，但是它为了在不同设计项目中能够得到应用，会重点强化其可移植性，因此它的设计代码规范更加严格。有的芯片公司专门从事IP核的开发和销售，ARM就是一个典型的例子，这些公司通过知识产权的授权营利<sup>[19]</sup>。

## 设计的辅助和自动化

---

由于集成电路系统的复杂性，工程师往往需要借助电子设计自动化工具来进行计算机辅助设计。逻辑综合就是电子设计自动化在数字集成电路设计中最显著的体现。以往在设计小规模、中规模集成电路时，工程师设计数字集成电路需要根据逻辑功能，通过类似卡诺图这样的手工途径来优化逻辑函数，然后确定使用何种逻辑门来实现电路。而在当前超大规模集成电路，乃至更大的甚大规模集成电路的设计中，这样的工作方式不太现实。电子设计自动化工具使得工程师能够从复杂的邏輯閘

设计转到功能设计，而底层的转换由自动工具完成，工程师只需要掌握如何设置这些工具工作策略的知识。硬件描述语言是集成电路设计自动化的重要基础。<sup>[8]:2-3</sup>电子设计自动化发展十分迅速，现在已经成立了诸如设计自动化会议的一些学术论坛，定期讨论业界的发展。<sup>[20]</sup>

完成整个集成电路设计常常涉及多个电子设计自动化工具的运用。<sup>[21]</sup>有些公司专门从事集成电路计算机辅助设计工具套件的开发和销售，例如Synopsys、Cadence、Mentor Graphics、Agilent、Altium、Xilinx等。电子设计自动化工具的本身作为一种软件，背后依靠的是各种计算机算法。因此电子设计自动化工具的开发更加接近软件设计的范畴，其开发人员需要重点关注逻辑简化、布局布线等方面的算法实现，但是他们同样需要了解集成电路的硬件知识。<sup>[5]:3-30</sup>

## 参考文献

---

1. Neil Weste, David Harris. CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition). Addison-Wesley. 1985. ISBN 978-0321547743.
2. 施敏. 半导体器件物理与工艺（第二版）. 苏州: 苏州大学出版社. ISBN 978-7-81090-015-7.
3. 沈理. SOC/ASIC设计、验证和测试方法学. 广州: 中山大学出版社. ISBN 7-306-02682-8.
4. 朱正涌, 张海洋, 朱元红. 半导体集成电路（第2版）. 北京: 清华大学出版社. ISBN 978-7-302-18512-3.
5. Andrew B. Kahng, Igor L. Markov, Jens Lienig, Jin Hu. VLSI Physical Design: From Graph Partitioning to Timing Closure. Springer. ISBN 978-90-481-9590-9.
6. Erik Erunvand. 数字VLSI芯片设计：使用Cadence和Synopsys CAD工具. 北京: 电子工业出版社. ISBN 978-7-121-09607-5.
7. 邓元庆、关宇、贾鹏、石会. 数字设计基础与应用. 北京: 清华大学出版社. ISBN 978-7-302-21406-9.
8. Samir Palnitkar. Verilog HDL: A Guide to Digital Design and Synthesis. 北京: 电子工业出版社. : 10. ISBN 7-121-00468-2.
9. Stephen Brown, Zvonko Vranesic. Fundamentals of Digital Logic with Verilog Design. McGraw-Hill Education. ISBN 0-07-283878-7.
10. Michael John, Sebastian Smith. 专用集成电路（英文名：Application Specific Integrated Circuits）. 北京: 电子工业出版社. ISBN 978-7-121-04036-8.
11. David Money Harris, Sarah L. Harris. 数字设计和计算机体系结构（英文名：Digital Design and Computer Architecture）. 北京: 机械工业出版社. ISBN 978-7-111-25459-1.
12. The Life of SPICE (PDF). Designers Guide. [2013-04-04]. （原始内容 (PDF)存档于2012-02-04）.
13. 虞希清. 专用集成电路设计实用教程. 浙江大学出版社. ISBN 978-7-308-05113-2.
14. Himanshu Bhatnagar. 高级ASIC芯片综合：使用Synopsys Design Compiler、Physical Compiler和PrimeTime. 北京: 清华大学出版社. ISBN 978-7-302-14881-4.
15. 袁媛, 谢巍, 刘明业. 基于高级综合的RTL综合对象及方法的研究. 北京理工大学学报. 2001, **21** (1).
16. 杨宗凯, 黄建, 杜旭. 数字专用集成电路的设计与验证. 电子工业出版社. : 125-138. ISBN 7-121-00378-3.
17. 克里斯·斯皮尔. SystemVerilog验证：测试平台编写指南（原书第二版）. 北京: 科学出版社. ISBN 978-7-03-025306-4.



18. Laung-Terng Wang, Cheng-Wen Wu and Xiaoqing Wen. VLSI Test Principles and Architectures: Design for Testability. Morgan Kaufmann. [ISBN 978-0123705976](#).
19. Michael Keating, Pierre Breacaud. 片上系统——可重用设计方法学（第二版）（英文名：Reuse Methodology Manual for System-on-a-Chip Designs, Third Edition）. 北京: 电子工业出版社. [ISBN 7-5053-9338-3](#).
20. [About Design Automation Conference \(DAC\)](#). Design Automation Conference. [2013-06-08]. （[原始内容存档于2013-06-14](#)）.
21. 池雅庆. ASIC芯片设计从实践到提高. 北京: 中国电力出版社. [ISBN 978-7-5083-5378-4](#).

## 参见

---

- [电路设计](#)、[电子设计自动化](#)、[计算机辅助设计](#)
- [电路](#)、[集成电路](#)、[超大规模集成电路](#)

---

取自“<https://zh.wikipedia.org/w/index.php?title=集成电路设计&oldid=65996034>”

---

本页面最后修订于2021年6月8日 (星期二) 21:11。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）

Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。

维基媒体基金会是按美国国内税收法501(c)(3)登记的非营利慈善机构。