

1.1 EDA 技术简介

信息社会的发展离不开集成电路，目前，集成电路正朝着速度快、容量大、体积小、功耗低的方向发展，实现这种进步的主要原因就是生产制造技术和电子设计技术的提升。前者以微细加工技术为代表，目前已进展到纳米级阶段，可以在几平方厘米的芯片上集成数亿个晶体管。后者的核心就是 EDA 技术，目前该技术已经渗透到电子产品设计的各个环节。

电子设计自动化（electronic design automation, EDA）是随着集成电路和计算机技术飞速发展应运而生的一种快速、有效、高级的电子设计自动化技术。EDA 工具融合了应用电子技术、计算机技术和智能化技术的最新成果，主要进行三个方面的辅助设计工作：集成电路（integrated circuit, IC）设计、电子电路设计及印制电路板（printed circuit board, PCB）设计。在数字设计领域，EDA 技术就是依靠功能强大的电子计算机和 EDA 软件工具，对以硬件描述语言（hardware description language, HDL）形式给出的系统设计文件自动地进行逻辑编译、化简、分割、综合、优化和仿真，直至下载到可编程逻辑器件（complex programmable logic device/field programmable gate array, CPLD/FPGA）或专用集成电路芯片中，实现既定的电路功能。EDA 技术使电路设计者的工作聚焦于利用硬件描述语言和 EDA 软件平台来完成对系统硬件功能的实现，极大地提高了设计效率，缩短了设计周期，节约了设计成本。

EDA 技术的应用范畴应包括电子工程师开发产品的全过程，以及电子产品生产过程中所有希望由计算机提供的各种辅助工作。从一个角度来看，EDA 技术可以粗略分为系统级、电路级和物理级三个层次的辅助设计过程；从另一个角度来看，EDA 技术应该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、从分立电路到集成电路的全部设计过程。EDA 技术的应用范畴如图 1-1 所示。

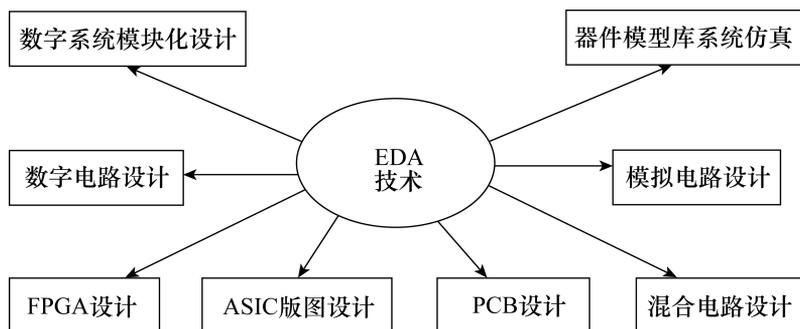


图 1-1 EDA 技术的应用范畴

1.2 EDA 技术及设计方法

1.2.1 EDA 技术特点

目前，电子技术的各个领域已全方位融入了 EDA 技术。IP 核已经在电子行业的产业、技术和设计领域得到了广泛应用，推动了整个行业的发展，使在 FPGA 上实现数字信号处理（digital signal processing, DSP）应用成为现实。同时，嵌入式微处理器软核的出现以及更大规模的可编程逻辑器件 CPLD/FPGA 的推出，使得可编辑片上系统（system on a programmable chip, SOPC）进入了大规模应用阶段。此外，支持标准硬件描述语言的 EDA 软件在仿真和设计两个方面也在不断更新。随着 EDA 技术的普及与深化，电子领域各学科的界限变得更加模糊，各学科相互包容和渗透的趋势愈加明显，EDA 技术在现代集成电路设计中扮演着至关重要的角色。它的特点主要体现在以下五个方面。

(1) 自动化程度高。EDA 工具可以自动完成许多复杂的设计任务，从而减少了人为错误，提高了设计效率和准确性。

(2) 可重复性强。在使用 EDA 工具进行设计时，能够确保每次生成的设计结果一致，从而提高了设计的可靠性。

(3) 高效性。EDA 工具能够大大缩短设计周期，加速产品的上市时间。

(4) 复杂性处理。EDA 工具可以处理极其复杂的电路设计，支持数百万甚至数十亿个晶体管的集成电路设计。

(5) 优化能力。EDA 工具提供各种优化功能，如功耗优化、面积优化和性能优化等，帮助设计者在不同的设计约束条件下找到最佳解决方案。

1.2.2 EAD 设计方法

在 EDA 技术普及之前，集成电路的设计主要依赖于手工设计方法。设计者需要手工绘制版图，这不仅增加了设计的复杂性和时间成本，还使得每一步都需要人工干预。此

外，由于缺乏有效的自动化工具以及手工设计的局限性，测试和验证阶段极为耗时费力，容易遗留设计错误，这些都使得整个设计过程更加困难。

EDA 设计方法引入了大量的自动化工具和流程，使得集成电路设计更加高效和可靠。其主要特点包括以下六个方面。

(1) 自动化的设计流程。EDA 工具集成了从设计到实现的整个过程，设计者可通过自动化工具实现电路的设计、验证和优化，提高设计效率和准确性。

(2) 设计语言支持。EDA 工具支持硬件描述语言（如 VHDL、Verilog HDL），设计者可以通过编写代码进行设计，增强了设计的灵活性和可移植性。

(3) 支持仿真和验证。EDA 工具提供强大的仿真和验证功能，可以在设计阶段检测和纠正错误，减少设计迭代次数。

(4) 综合和布局布线。EDA 工具能够自动进行逻辑综合和物理布局布线，提高了设计效率和质量。

(5) 多层次优化。EDA 工具能够在不同层次上进行优化，包括逻辑层、物理层和系统层，确保设计在性能、功耗和面积等方面达到最佳平衡。

(6) 支持协同设计。EDA 工具支持协同设计，多个设计团队可以同时不同模块的设计和集成，提高了设计效率。

总的来说，EDA 技术通过高度的自动化和优化能力，大大提升了集成电路设计的效率和质量，推动了半导体行业的快速发展。

IEEE 标准的 HDL（如 VHDL、Verilog HDL）给 PLD 和数字系统的设计带来了更新的设计方法和理念，产生了目前最常用的、被称为“自顶向下”（top-down）的设计方法。“自顶向下”的设计采用功能分割的方法，从顶向下逐次将设计内容进行分块和细化。在设计过程中采用层次化和模块化方法使系统设计变得简洁和方便。层次化设计是分层次、分模块地进行设计描述。描述器件总功能的模块放在最上层，称为顶层设计；描述器件某一部分功能的模块放在下层，称为底层设计。底层模块还可以再向下分层，直至最后完成硬件电子系统电路的整体设计。

PLD 的出现给数字系统的传统设计带来了新的变革。采用 PLD 进行的数字系统设计，是基于芯片的设计或被称为“自底向上”（bottom-up）的设计，它与传统的积木式设计有本质的不同。自底向上的设计可以直接通过设计 PLD 芯片来实现数字系统功能，将原来由电路板设计完成的大部分工作放在 PLD 芯片的设计中进行。采用这种新的设计方法时，设计者可以根据实际情况和要求定义器件的内部逻辑关系和管脚，这样可以通过芯片设计实现多种数字系统功能。同时，由于管脚定义的灵活性，不但大大减轻了系统设计的工作量和难度，提高了工作效率，而且还可以减少芯片数量，缩小系统体积，降低能源消耗，提高系统的稳定性和可靠性。如图 1-2 所示为电子系统的“自顶向下”和“自底向上”的设计步骤。

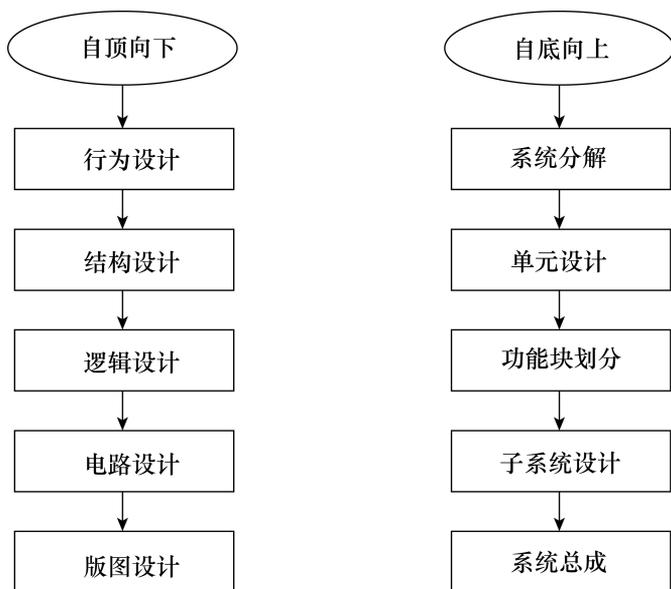


图 1-2 “自顶向下”与“自底向上”的设计步骤

“自顶向下”是一种概念驱动的设计方法。该方法要求设计者在整个设计过程中尽量运用概念去描述和分析设计对象，而不要过早地考虑实现该设计的具体电路、元器件和工艺，以便抓住主要矛盾，避免纠缠在具体细节上，这样才能控制住设计的复杂性。整个设计在概念上的演化从顶层到底层应当逐步由概括到展开、由粗略到精细。只有当整个设计在概念上得到验证与优化后，才能考虑“采用什么电路、元器件和工艺去实现该设计”这类具体问题。在进行“自顶向下”的设计时，从系统级设计入手，在顶层进行功能方框图的划分和结构设计；在方框图一级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述；在功能一级进行验证，然后用逻辑综合优化工具生成具体的门级逻辑电路网表，其对应的物理实现级可以是印制电路板或专用集成电路。而“自底向上”的设计方法一般是在系统划分和分解的基础上先进行单元设计，在完成单元设计后逐步向上进行功能模块设计，然后再进行子系统的设计，最后完成系统的总成设计。“自顶向下”的设计方法有利于在早期发现结构设计中的错误，提高设计的成功率，因而在现代 EDA 系统中被广泛采用。如图 1-3 所示为“自底向上”设计框架。

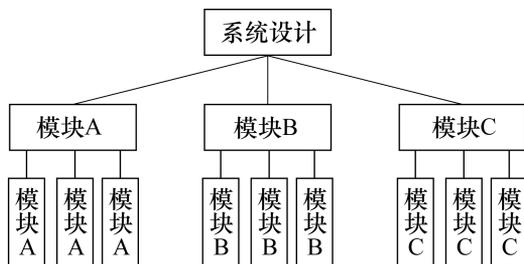


图 1-3 “自底向上”设计框架

1.3 EDA 设计流程

利用 EDA 技术进行电路设计的大部分工作是在 EDA 软件平台上进行的。一个典型的 EDA 设计流程主要包括设计准备、设计输入、设计处理、设计校验、器件编程和器件测试与验证等六个基本步骤，如图 1-4 所示。

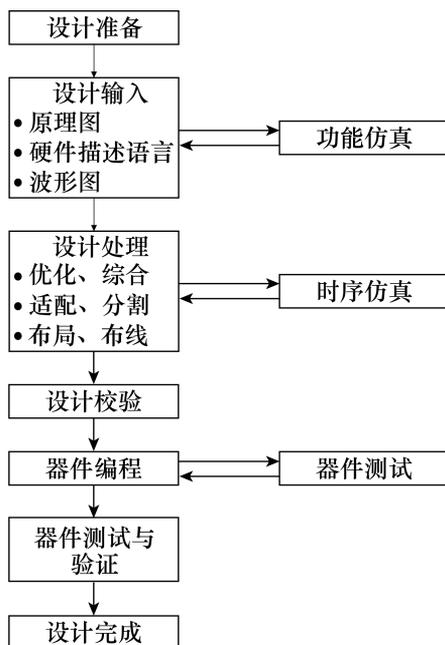


图 1-4 EDA 设计流程

1.3.1 设计准备

设计准备是指设计者在进行设计之前，依据任务要求，确定系统要完成的功能及复杂程度，器件资源的利用、成本等所要做的准备工作，如进行方案论证、系统设计和器件选择等。

1.3.2 设计输入

设计输入是将设计的系统或电路用开发软件要求的某种形式表示出来，并送入计算机的过程。常用的设计输入方式包括文本方式设计输入方式、图形设计输入方式和混合设计输入方式三种，具体特点如下。

(1) 原理图或图形输入方式。这是一种最直接的设计输入方式，它使用软件系统提供的元器件库及各种符号和连线画出原理图，形成原理图输入文件。通常在对系统及各部分电路很熟悉时使用，或用于系统对时间特性要求较高的场合。优点是容易实现仿真，便于信号的观察和电路的调整。

(2) 硬件描述语言输入方式。硬件描述语言包括普通硬件描述语言和行为描述语言, 它们都是用文本方式描述设计和输入。普通硬件描述语言有 AHDL、CUPL 等, 它们支持逻辑方程、真值表、状态机等逻辑表达方式。而行为描述语言是目前常用的高层硬件描述语言, 有 VHDL 和 Verilog HDL 等, 它们具有很强的逻辑描述和仿真功能, 可实现与工艺无关的编程与设计, 使设计者在系统设计、逻辑验证阶段便可确定方案的可行性, 而且行为描述语言的输入效率高, 方便设计者在不同的设计输入库之间转换。总之, 运用 VHDL、Verilog HDL 硬件描述语言进行设计已是当前的趋势。

(3) 波形输入方式。波形输入主要用于建立和编辑波形设计文件、输入仿真向量和功能测试向量。波形设计输入适用于时序逻辑和有重复性的逻辑函数, 系统软件可以根据用户定义的输入/输出波形自动生成逻辑关系。同时, 波形编辑功能还允许设计者对波形进行拷贝、剪切、粘贴、重复与伸展, 从而可以用内部节点、触发器和状态机建立设计文件, 并将波形进行组合, 显示各种进制的状态值, 还可以通过将一组波形重叠到另一组波形上, 对两组仿真结果进行比较。

1.3.3 设计处理

设计处理是 EDA 设计的核心环节。在设计处理阶段, 编译软件将对设计输入文件进行逻辑化简、综合和优化, 并适当地用一片或多片器件自动地进行适配, 最后产生编程用的编程文件。设计处理主要包括设计编译和检查、逻辑优化和综合、适配和分割、布局和布线、生成编程数据文件等过程。

(1) 设计编译和检查。设计输入完成之后, 立即进行编译。在编译过程中首先进行语法检验, 如检查原理图的信号线有无漏接, 信号有无双重来源, 文本输入文件中的关键字有无错误等, 并及时标出、报告错误的位置信息, 供设计者修改。然后进行设计规则检验, 检查总的设计有无超出器件资源或规定的限制并列出生成报告, 指明违规之处和电路的潜在不可靠情况以供设计者纠正。

(2) 逻辑优化和综合。逻辑优化是化简所有的逻辑方程或用户自建的宏, 使设计所占用的资源最少。综合的目的是将多个模块化设计文件合并为一个网表文件, 并使层次设计平面化(即展平); 逻辑综合是 20 世纪 90 年代电子学领域兴起的一种新的设计方法, 是以系统级设计为核心的高层次设计。这种方法是将最新的算法与工程界多年积累的设计经验结合起来, 自动地将用真值表、状态图以及 VHDL 硬件描述语言等所描述的数字系统转化为满足设计性能指标要求的逻辑电路, 并对电路进行速度、面积等方面的优化。其主要过程包括: 逻辑结构的生成与优化、逻辑网络的性能优化。前者主要是进行逻辑化简与优化, 尽可能用较少的元器件和连线形成一个逻辑网络结构(逻辑图), 满足系统逻辑功能的要求; 后者是利用给定的逻辑单元库, 对已生成的逻辑网络进行元器件配置, 从而估算实现该逻辑网络的芯片的性能与成本。性能主要指芯片的速度, 成本主要指芯片的面积与功耗。速度与面积或速度与功耗是矛盾的。这里允许使用者对速度与面积或速度与功耗相矛盾的指标进行性能与成本的折中, 以确定合适的元器件配置, 完成最终的、符合要求的逻辑网络结构。

(3) 适配和分割。在适配和分割过程中, 确定优化以后的逻辑能否与下载目标器件

CPLD 或 FPGA 中的宏单元和 I/O 单元适配，然后将设计分割为多个便于适配的逻辑小块形式并映射到器件相应的宏单元中。当整个设计不能装入一片器件时，则将整个设计自动分割成多块并装入同一系列的多片器件中。

(4) 布局和布线。布局和布线工作是在设计检验通过以后由软件自动完成的，它能以最优的方式对逻辑元件布局，并准确地实现元件间的布线互连。完成布局和布线以后，软件会自动生成布线报告，并提供有关设计中各部分资源的使用情况等信息。

(5) 生成编程数据文件 (JED 文件)。设计处理的最后一步是产生可供器件编程使用的数据文件。对 CPLD 来说，是产生熔丝图文件 (电子器件工程联合会制定的标准格式文件，简称 JED 文件)；对 FPGA 来说，是生成位流数据文件 (bit-stream generation，简称 BSG 文件)。

1.3.4 设计校验

设计校验包括功能仿真和时序仿真，这两项工作通常在设计处理过程中同时进行。功能仿真是在设计输入完成之后，选择具体器件进行编译之前的逻辑功能验证，因此又称为前仿真。此时的仿真没有延时信息或者只有由系统添加的微小标准延时，这对于初步的功能检测非常方便。而时序仿真是在选择了具体器件并完成布局、布线之后进行的时序关系仿真，因此又称为后仿真或延时仿真。在设计处理后的设计校验，是对系统和各模块进行时序仿真，分析其时序关系，估计设计的性能以及检查和消除竞争冒险等。

1.3.5 器件编程

器件编程是指将设计处理中产生的编程数据文件通过软件放到具体的可编程逻辑器件中。对 CPLD 器件来说，是将 JED 文件下载到 CPLD 器件中；对 FPGA 来说，是将位流数据 BSG 文件配置到 FPGA 中。器件编程需要满足一定的条件，包括编程电压、编程时序和编程算法等。普通的 CPLD 器件和一次性编程的 FPGA 需要使用专用的编程器完成器件的编程工作。基于静态随机存取存储器 (static random access memory, SRAM) 的 FPGA 可以由可擦除可编程只读存储器 (erasable programmable read only memory, EPROM) 或其他存储器进行配置。在系统的可编程逻辑器件 (ISP-PLD) 则不需要专门的编程器，只要一根与计算机互连的下载编程电缆即可。

1.3.6 器件测试与验证

在编程完毕之后，可以使用编译时产生的文件对器件进行检验、加密等工作，或采用边界扫描测试技术进行功能测试，测试成功后才算设计完成。

设计验证可以在 EDA 硬件开发平台上进行。EDA 硬件开发平台的核心部件是一片可编程逻辑器件 FPGA 或 CPLD，再附加一些输入输出设备，如按键、数码显示器、指示灯、喇叭等，还提供时序电路需要的脉冲源。具体过程是先将设计电路编程下载到 FPGA 或 CPLD 中，根据 EDA 硬件开发平台的操作模式要求，进行相应的输入操作，然后检查输出结果，验证设计电路。

1.4 EDA 技术的发展

20 世纪末，数字电子技术得到飞速发展，有力地推动了社会生产力的发展和社会信息化的提高。如今，数字电子技术的应用已经渗透到人类生活的各个方面。从计算机到手机，从数字电话到数字电视，从家用电器到军用设备，从工业自动化到航天技术，都广泛应用了数字电子技术。

微电子技术，即大规模集成电路加工技术的进步是现代数字电子技术发展的基础。目前，在硅片单位面积上集成的晶体管数量越来越多，1978 年推出的 8086 微处理器芯片集成的晶体管数是 4 万只，到 2000 年推出的 Pentium 4 微处理器芯片的集成度达 4200 万只晶体管，2005 年生产 PLD 的集成度达到 5 亿只晶体管，包含的逻辑元件 LEs (logic elements) 有 18 万个，2006 年生产的 PLD 中的 LEs 达到 33 万个，而 2009 年生产的 PLD 中的 LEs 达到 84 万个，集成度达到 25 亿只晶体管。原来需要成千上万只电子元件组成的一台计算机主板或彩色电视机电路，而现在仅用几片超大规模集成电路就可以代替，现代集成电路已经能够实现单片电子系统 (system on a chip, SOC) 的功能。

现代电子设计技术的核心是电子设计自动化 (electronic design automation, EDA) 技术。它依靠功能强大的电子计算机，在 EDA 工具软件平台上对以 HDL 为系统逻辑描述手段完成的设计文件自动地完成逻辑编译、化简、分割、综合、优化、仿真，直至下载到可编程逻辑器件 CPLD/FPGA 或专用集成电路 (application specific integrated circuit, ASIC) 芯片中，实现既定的电子电路设计功能。总之，EDA 技术使得电子电路设计者的工作仅限于利用硬件描述语言和 EDA 软件平台来完成对系统硬件功能的实现，极大地提高了芯片设计效率，减少设计周期，节省设计成本。

EDA 是 20 世纪 90 年代初在从计算机辅助设计 (computer aided design, CAD)、计算机辅助制造 (computer aided manufacturing, CAM)、计算机辅助测试 (computer aided testing, CAT) 和计算机辅助工程 (Computer Aided Engineering, CAE) 的基础上发展而来的。一般把 EDA 技术的发展分为 CAD、CAE 和 EDA 三个阶段。

(1) CAD 是 EDA 技术发展的早期阶段。在这个阶段，人们开始利用计算机取代手工劳动。但当时的计算机硬件功能有限，软件功能较弱，人们主要借助计算机对设计的电路进行一些模拟和预测，辅助进行集成电路版图编辑、PCB 布局布线等简单的版图绘制等工作。

(2) CAE 是在 CAD 工具逐步完善的基础上发展起来的，在设计方法学、设计工具集成化方面取得了长足的进步以后，人们可以利用计算机作为单点设计工具，并建立各种设计单元库，开始用计算机将许多单点工具集成在一起使用，大大提高了工作效率。

(3) 20 世纪 90 年代以来，微电子工艺有了惊人的发展，工艺水平已经达到 45 nm 级，在一个芯片上已经可以集成上百万乃至数亿只晶体管，芯片速度达到了吉比特/秒量级，百万门以上的可编程逻辑器件陆续面世，这样就对电子设计的工具提出了更高的要求，也为电子设计提供了更广阔的发展空间，促进了 EDA 技术的形成。世界上各 EDA 公

司推出了兼容各种硬件实现方案和支持标准硬件描述语言的 EDA 工具软件，有效地将 EDA 技术推向成熟。

如今 EDA 技术已经成为电子设计的重要工具，无论是设计芯片还是设计系统，如果没有 EDA 工具的支持都将是难以完成的。EDA 工具已经成为现代电路设计师的重要武器，起到越来越重要的作用。

随着市场需求的增长，集成工艺水平及计算机自动设计技术的不断提高，促进单片系统，或称系统集成芯片成为 IC 设计的发展方向。由于以往的 HDL 语言只提供行为级或功能级的描述，无法完成更复杂的系统级的抽象描述，人们开始尝试开发一些新的系统级设计语言，现在已开发出更趋于电路系统行为级的硬件描述语言，如 System Verilog、System C 及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 HDL 语言（Verilog HDL、VHDL）或其他更低层次描述模块的混合仿真。多家公司商用的 C/C++ 综合器已经趋向成熟，可以直接把 C/C++ 程序转换为 HDL。虽然用户用高级语言编写的模块只能部分自动转化成 HDL 描述，但作为一种针对特定应用领域的开发工具，软件供应商已经为常用的功能模块提供了丰富的宏单元库支持，可以方便地构建应用系统，并通过仿真加以优化，最后自动产生 HDL 代码，进入下一阶段的 ASIC 实现。

此外，随着系统开发对 EDA 技术的目标器件各种性能要求的提高，ASIC 和 FPGA 将更大程度地相互融合。这是因为虽然标准逻辑 ASIC 芯片尺寸小、功能强大、功耗低，但设计复杂，并且有批量生产要求；而可编程逻辑器件开发费用低，能在现场进行编程，但其体积大、功能有限，而且功耗较大。因此，FPGA 和 ASIC 正在互相融合，取长补短。由于一些 ASIC 制造商提供具有可编程逻辑的标准单元，可编程器件制造商重新对标准逻辑单元产生兴趣，而有些公司采取两头并进的方法，从而使市场开始发生变化，在 FPGA 和 ASIC 之间正在诞生一种“杂交”产品，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。

尽管将标准单元核与可编程器件集成在一起并不意味着使 ASIC 更加便宜或使 FPGA 降低功耗，但是这种操作可使设计者将两者的优点结合在一起，通过去掉 FPGA 的一些功能，可减少成本和开发时间并增加灵活性。当然，现如今也在进行将 ASIC 嵌入可编程逻辑单元的工作。目前，许多 PLD 公司开始为 ASIC 提供 FPGA 内核。PLD 厂商与 ASIC 制造商结盟，为 SOC 设计提供嵌入式 FPGA 模块，使未来的 ASIC 供应商有机会更快地进入市场，利用嵌入式内核获得更长的市场生命期。例如，在实际应用中使用 SOC FPGA，即将 FPGA 内核与嵌入式 RISC 微控制器组合在一起形成新的 SOC，广泛用于电信、网络、仪器仪表和汽车中的低功耗应用系统中。当然，也有 PLD 厂商不把 CPU 的硬核直接嵌入 FPGA 中，而是使用软 IP 核即 SOPC，也可以完成复杂电子系统的设计，只是成本将相应提高。

另外，现在传统 ASIC 和 FPGA 之间的界限正变得模糊。系统级芯片不仅集成 RAM 和微处理器（ARM、x86），也集成 FPGA。整个 EDA 和 IC 设计工业都在朝这个方向发展，这并非 FPGA 与 ASIC 制造商竞争的产物。对于用户来说，这意味着有了更多的选择，“深度学习”的应用使得 GPU+CPU 架构的异构计算获得蓬勃发展，也推动了 FPGA+CPU 架构的发

展。相比于前者，FPGA+CPU 的功耗更低，系统灵活性更强，也比较容易向 ASIC 转化。

总之，随着 EDA 工具能力的提升和 FPGA 规模的增加，并行计算能力在 FPGA 应用中表现突出，因而高速图像处理、人工智能、数据中心、云计算、高速接口、存储中心的架构方案中越来越多地使用 FPGA。

1.5 EDA 常用工具

为了完成复杂的 ASIC 设计，EDA 系统一般包括 10~20 个 CAD 工具。现在 EDA 技术和系统设计工具正逐渐被理解成一个整体的概念，即电子系统设计自动化（electronic system design automatic, ESDA）。在过去 30 多年中，人们开发了大量 CAD 工具来完成集成电路的设计，主要的 EDA 设计工具如图 1-5 所示，包括检查/分析工具、编辑器、仿真器、优化/综合工具等。

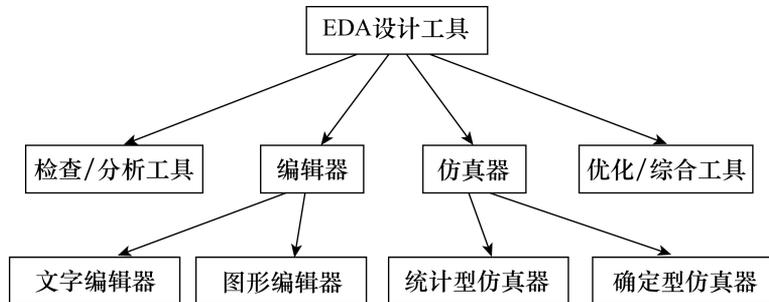


图 1-5 EDA 设计工具

通常专业的 EDA 工具供应商或可编程逻辑器件厂商都提供 EDA 开发工具，在这些 EDA 开发工具中都含有设计输入编辑器，如 Xilinx 公司的 Foundation、Altera 公司的 Quartus II 等。仿真与验证 EDA 工具见表 1-1，综合 EDA 工具见表 1-2。

表 1-1 仿真与验证 EDA 工具

设计周期	工具分类	工具名称	供应商
仿真与验证	Digital Simulator	NC-Sim/Verilog-XL	Cadence
		VCS	Synopsys
	Equivalence Check	Encounter Conformal Equivalence Checker	Cadence
		Formality、ESP	Synopsys
		FormalPro	Mentor
	Waveform Viewer	Debussy、Verdi	Novas

续表

设计周期	工具分类	工具名称	供应商
仿真与验证	Analog Simulator	HSpice、Nanosim	Synopsys
		Incisive AMS	Cadence
		Advance MS	Mentor
	RTL Code Coverage	HDL Score	Innoveda
		VCS	Synopsys
		VN-Cover	TransEDA
	RTL Syntax and SRS Checker	nLint	Novas
		SpyGlass	Atrenta
		Leda	Synopsys
	C++ Based System Testbench	Nucleus C++	Mentor

表 1-2 综合 EDA 工具

设计周期	工具分类	工具名称	供应商
综合	Clock Gating	Power Compiler	Synopsys
		Power-savvy	Azuro
	RTL Synthesis	Design Compiler	Synopsys
	Physical Synthesis	Blast Fusion	Magma
		Physical Compiler	Synopsys

在 EDA 技术中，仿真的地位非常重要。行为模型的表达、电子系统的建模、逻辑电路的验证以及门级系统的测试，每一步都离不开仿真器的模拟检测。逻辑电路及各个环节的快速仿真是整个 EDA 设计流程中最重要、最耗时的一个步骤。因此，仿真器的仿真速度、仿真的准确性和易用性成为衡量仿真器性能的重要指标。几乎每个 EDA 厂商都提供基于 Verilog HDL/VHDL 的仿真器。常用的仿真器有 Model Technology 公司的 Modelsim，Cadence 公司的 Verilog-XL 和 NC-Sim、Aldec 公司的 Active HDL、Synopsys 公司的 VCS 等。

同时，HDL 综合器是一种用 EDA 技术实施电路设计中完成电路化简、算法优化、硬件结构细化的计算机软件，是将硬件描述语言转化为硬件电路的重要工具。HDL 综合器在把可综合的 HDL（Verilog HDL 或 VHDL）转化为硬件电路时，要经过两个步骤。第一步是 HDL 综合器对 Verilog HDL 或 VHDL 进行处理分析，并将其转换成电路结构或模块，这是不考虑实际器件实现的，即完全与硬件无关。这个过程是一个通用电路原理图形成的过程。第二步是对实际实现目标器件的结构进行优化，并使之满足各种约束条件，优化关键路径等。

HDL 综合器的输出文件一般是网表文件，这是一种用于电路设计数据交换和交流的工业标准化格式的文件，可以直接用硬件描述语言 HDL 表达的标准格式的网表文件，也可以对应 FPGA/CPLD 器件厂商的网表文件。HDL 综合器是 EDA 设计流程中的一个独立的设计步骤，它常在其他 EDA 环节中被调用，以完成整个设计流程。生成的电路网表（逻辑图）由元件名 N、模型 M、输入端信号 PI、输出端信号 PO 四部分组成，是唯一确定电路连接关系的数据结构，即 $E = (N, M, PI, PO)$ 。

例如：1 位全加器的电路网表。

$E_1, \text{ xor}, (X, Y), S_1$

$E_2, \text{ xor}, (C_{in}, S_1), SUM$

$E_3, \text{ and}, (X, Y), S_2$

$E_4, \text{ and}, (S_1, C_{in}), S_3$

$E_5, \text{ or}, (S_2, S_3), C_{out}$

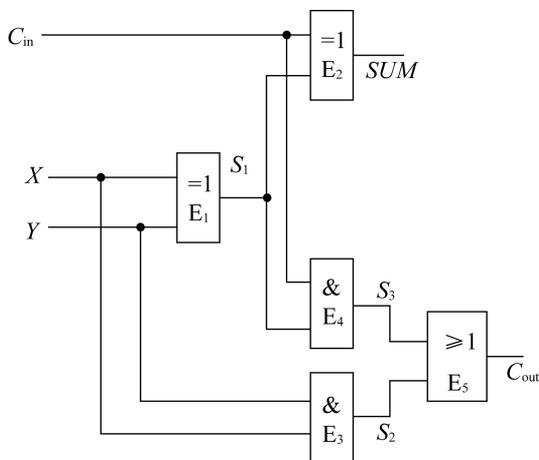


图 1-6 1 位全加器的电路

1.6 Quartus II 简介

由于本书给出的实验和设计是基于 Quartus II 开展的，其应用方法和设计流程对于其他流行的 EDA 工具而言具有一定的典型性和一般性，所以在此对 Quartus II 做一些介绍。

Quartus II 是 Altera 公司提供的 FPGA/CPLD 开发集成环境。它是 Altera 公司在 21 世纪初推出的，其前一代产品是 FPGA/CPLD 集成开发环境 MAX+Plus II。Quartus II 提供了完整的多平台设计环境，能满足各种特定设计的需要，并为单芯片 SOPC 和 Altera DSP 开发包进行系统模型设计提供了集成综合环境。其设计工具完全支持 Verilog HDL、VHDL 的设计流程，且内部嵌有 Verilog HDL、VHDL 逻辑综合器。Quartus II 也可以兼容第三方的综合工具，如 Leonardo Spectrum、Synplify Pro、DC-FPGA 等，并能直接调用这些工具。同

时，它也具备仿真功能，支持第三方的仿真工具，如 ModelSim。此外，还可以与 Matlab 和 DSP Builder 结合，进行基于 FPGA 的 DSP 系统开发，是 DSP 硬件系统实现的关键 EDA 工具。Quartus II 包括模块化的编译器，主要包括分析/综合器（analysis & synthesis）、适配器（fitter）、装配器（assembler）、时序分析器（timing analyzer）、设计辅助模块（design assistant）、EDA 网表文件生成器（EDA netlist writer）、编辑数据接口（compiler database interface）等功能模块。

图 1-7 给出了 Quartus II 的编译流程，上排所示为 Quartus II 编译设计的主控界面，显示了 Quartus II 自动设计的各主要处理环节和设计流程，包括图形与 HDL 编辑器、分析与综合器、适配器、编程文件装配器、编程器五个器件。下排是与上面的 Quartus II 设计流程相对照的标准的 EDA 开发流程。

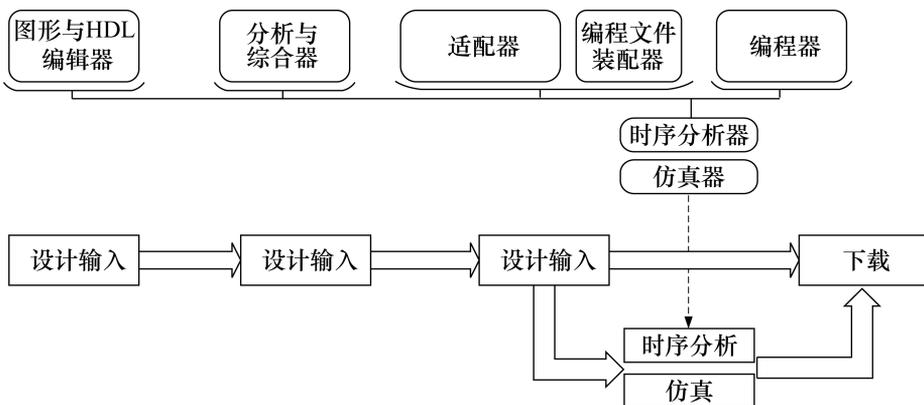


图 1-7 Quartus II 编译流程

本章主要介绍可编程逻辑器件 PLD 的概念、分类、工作原理、发展趋势和测试技术，并详细介绍了 CPLD 和 FPGA 两种可编程逻辑器件的电路结构、工作原理和使用方法。

2.1 PLD 概述

2.1.1 PLD 的发展历程

PLD 是一种半定制集成电路，其内部集成了大量的门和触发器等基本逻辑电路，用户通过编程改变 PLD 的内部逻辑关系或连线，就可以得到所需的电路。PLD 的出现，改变了传统的数字系统设计方法。

在 PLD 出现之前，数字系统的传统设计往往采用“积木”式的方法进行，通过将标准集成电路器件搭建成电路板来实现系统功能，即先由器件搭成电路板，再由电路板搭成系统。

PLD 的主要发展历程包括六个阶段。20 世纪 70 年代初，是通过熔丝编程的可编程只读存储器（programmable read-only memory, PROM）和可编程逻辑阵列（programmable logic array, PLA）。20 世纪 70 年代末，AMD 公司推出可编程阵列逻辑（programmable array logic, PAL）。20 世纪 80 年代初，Lattice 公司发明了通用阵列逻辑（generic array logic, GAL）。20 世纪 80 年代中期，Xilinx 公司生产出了世界上第一个现场可编程门阵列器件（field programmable gate array, FPGA）。同一时期，Altera 公司推出了电可编程逻辑器件（electrically programmable logic device, EPLD）。20 世纪 80 年代末，Lattice 公司又提出了在系统可编程技术（in-system programmability, ISP），并且推出了一系列具备在系统可编程能力的复杂可编程逻辑器件（complex programmable logic device, CPLD）。20 世纪 90 年代以后，可编程逻辑器件的规模超过了百万逻辑门，并且出现了内嵌复杂功能块（如加法器、乘法器、RAM、PLL CPU 核、DSP 核等）的超大规模器件 SOPC。

目前常用的可编程逻辑器件都是从与或阵列和门阵列发展起来的，所以，以结构分类，PLD 分为阵列型 PLD 和现场可编程门阵列型 FPGA 两大类。阵列型 PLD 的基本结构由输入缓冲电路、输出缓冲电路、与阵列、或阵列组成，如图 2-1 所示。

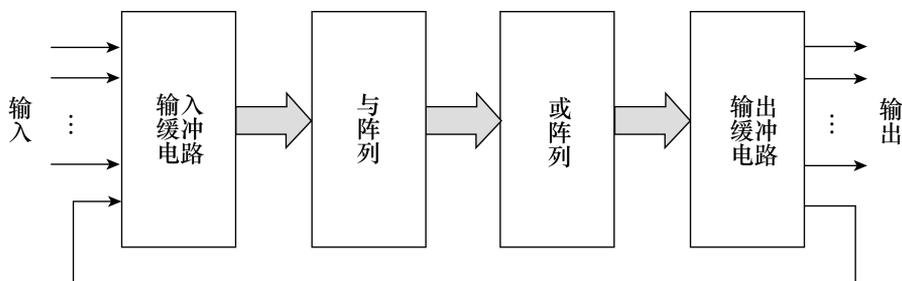


图 2-1 阵列型 PLD 器件的基本结构

目前，依据可编辑的部位可将 PLD 器件分为 PROM、PLA、PAL、GAL 四种最基本的类型，各自的特点如下。

(1) PROM。PROM 除用作只读存储器外，还可作为 PLD 使用。两个 PROM 器件主要由地址译码部分、PROM 单元阵列和输出缓冲部分构成。图 2-2 是 PROM 的基本结构，也可从 PLD 的角度来分析 PROM 的基本结构。PROM 中的地址译码器完成的是对 PROM 存储阵列的行的选择，这样可以将 PROM 的地址译码器看成一个与阵列，而将存储单元阵列看成一个或阵列，其中或阵列可编程，与阵列不可编程。因此，PROM 的逻辑阵列结构如图 2-3 所示。

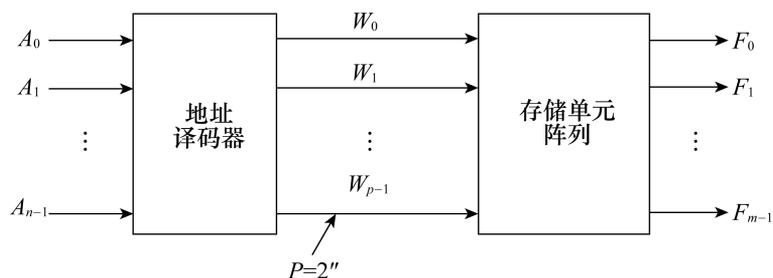


图 2-2 PROM 的基本结构

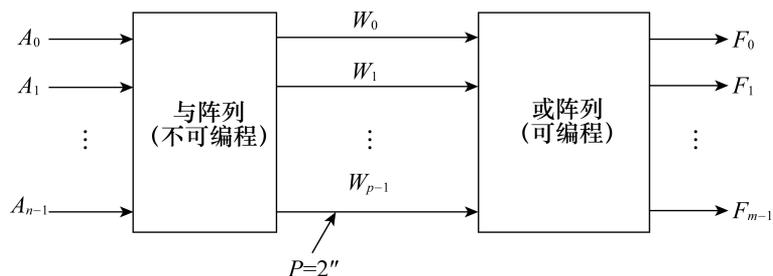


图 2-3 PROM 的逻辑阵列结构

(2) PLA。PLA 主要由与逻辑阵列和或逻辑阵列两个部分组成。其中，与逻辑阵列作为输入矩阵，接收输入信号，并通过与门进行组合，形成各种逻辑乘积项；或逻辑阵列作为输出矩阵，接收与逻辑阵列的输出信号，并通过或门进行汇总，最终形成输出信号。在 PLA 中，与、或阵列之间的连接由编程开关矩阵控制，通过编程设置开关的状态来实现不同的逻辑功能。其基本结构和逻辑阵列结构分别如图 2-4、图 2-5 所示。

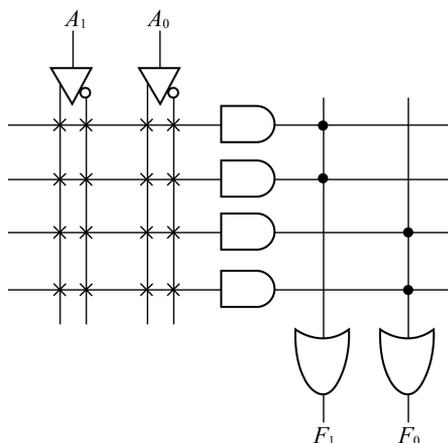


图 2-4 PLA 的基本结构

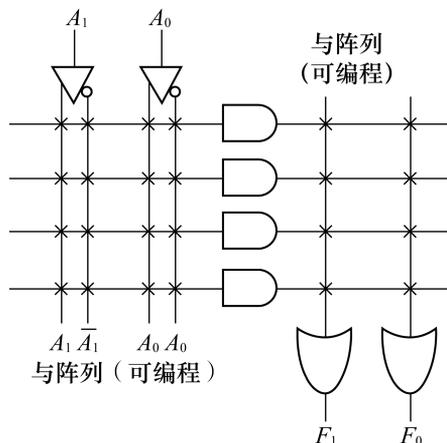


图 2-5 PLA 的逻辑阵列结构

(3) PAL。PAL 主要由可编程与阵列、固定或阵列以及可编程输出电路组成。可编程的与阵列由多个与门组成，其基本结构如图 2-6 所示。这些与门用来接收来自输入引脚或内部信号的多个信号，并执行与运算。而固定的或阵列由多个或门组成，这些或门接收来自与阵列的乘积项作为输入，并执行或运算以产生最终的输出信号。相对于与阵列的灵活性，或阵列的结构是固定的，但它允许用户选择与哪些乘积项相连，来定制相应的输出逻辑。

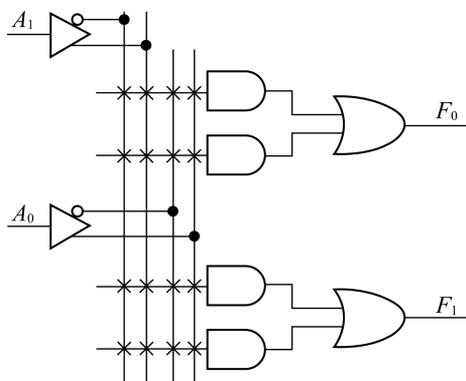


图 2-6 PAL 的基本结构

PAL 器件还包括可编程的输出电路，这些电路通常包括三态门 (tri-state buffers) 和可能的反馈机制。三态门允许用户控制输出信号的激活状态，以及是否将信号传递到外部引脚。在某些情况下，输出电路还可以包括触发器 (如 D 触发器)，用于在时钟信号控制下存储或阵列的输出结果，并将存储结果作为最终输出信号。

(4) GAL。GAL 的基本结构如图 2-7 所示，主要由可编程与逻辑阵列、输出逻辑宏单元以及输入输出缓冲器等部分组成。其中，可编程与逻辑阵列是 GAL 的核心部分，用于实现输入信号的逻辑与操作。同时，GAL 在每个输出端都集成一个输出逻辑宏单元，用于定义每个输出的结构和功能。输出逻辑宏单元可以根据需要配置为不同的逻辑组态，如专用输入方式、组合型输出方式以及寄存器型输出方式等。GAL 中的输入缓冲器用于接收外部输入信号，并将其转换为适合内部电路处理的信号；而输出缓冲器用于将内部电路处理后的信号输出到外部电路中，提供必要的驱动能力和信号完整性。部分 GAL 器件还包含反馈/输入缓冲器、系统时钟和控制信号，它们用于实现输出信号的反馈或作为额外的输入信号源、系统时钟和控制信号来同步内部操作或控制输出状态。

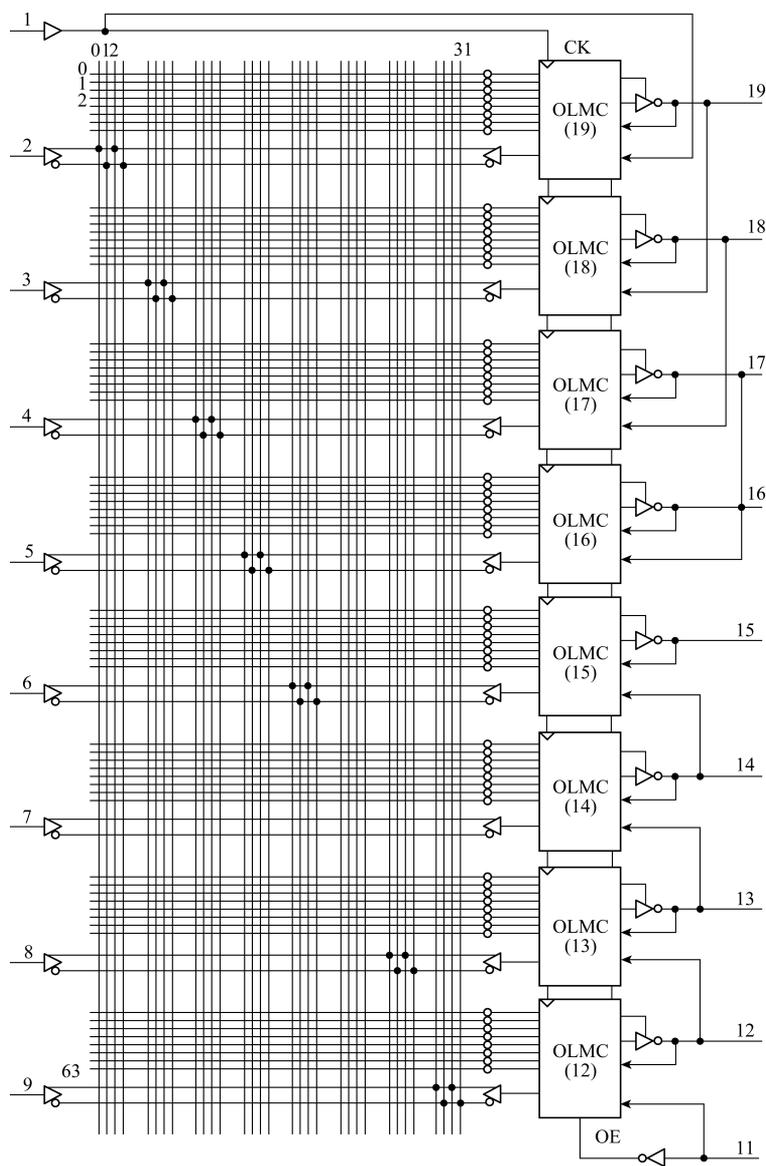


图 2-7 GAL 的基本结构

总之，与/或阵列是 PLD 器件中最基本的结构，通过编程改变与阵列和或阵列的内部连接，可以实现不同的逻辑功能。PROM、PLA、PAL 和 GAL 四种 PLD 器件的比较见表 2-1。

表 2-1 四种 PLD 器件的比较

器件名	与阵列	或阵列	输出电路
PROM	固定	可编程	固定
PLA	可编程	可编程	固定
PAL	可编程	固定	固定
GAL	可编程	固定	可组态

随着技术的进步，PLD 也逐渐向系统内可重构的方向发展，使设计者能够在系统运行过程中灵活地修改和调整逻辑功能，增强了系统的灵活性和适应性。此外，为了适应现代电子设备的要求，PLD 也在不断优化其功耗特性，向着低电压、低功耗的方向发展。这种趋势有助于延长设备的使用时间，提高能效，并减少对环境的影响。另外，高速电子系统的普及对逻辑器件的延时特性也提出了更高的要求。因此，PLD 正朝着高速可预测延时器件的方向发展，以满足对时序控制精度要求极高的应用场景。为了应对多样化的设计需求，PLD 技术还在向混合可编程技术方向发展，即将多种可编程技术（如 FPGA、CPLD、PLA 等）融合在一起，形成更强大的可编程逻辑平台。这种混合可编程技术不仅能够提供更灵活的设计选项，还能够充分发挥各种技术的优势，满足各种复杂应用场景的需求。

综上所述，PLD 技术的发展趋势涵盖了高密度、大规模化、系统内可重构、低电压低功耗、高速可预测延时以及混合可编程技术等多个方面，这些趋势将共同推动 PLD 技术在未来的广泛应用和持续发展。

2.1.2 PLD 的分类

1. 按集成密度分类

目前，可编程逻辑器件种类繁多。从集成密度上可分为低密度可编程逻辑器件（low-density PLD, LDPLD）和高密度可编程逻辑器件（high-density PLD, HDPLD）两类，如图 2-8 所示。PROM、PLA、PAL 和 GAL 是早期发展起来的 PLD，其集成密度一般小于 700 门（等效门）/片，它们同属于 LDPLD。而 HDPLD 包括可擦除可编程逻辑器件（erasable PLD, EPLD）、复杂可编程逻辑器件（complex PLD, CPLD）和 FPGA 三种，其集成密度大于 700 门/片。随着集成工艺的发展，HDPLD 的集成密度不断增加，性能不断提高。例如 Altera 公司的 EPM9560，其密度为 12 000 门/片，Lattice 公司的 pLSI/ispLSI3320，其密度为 14 000 门/片等。目前，集成度最高的 HDPLD 可达 25 万门/片以上。

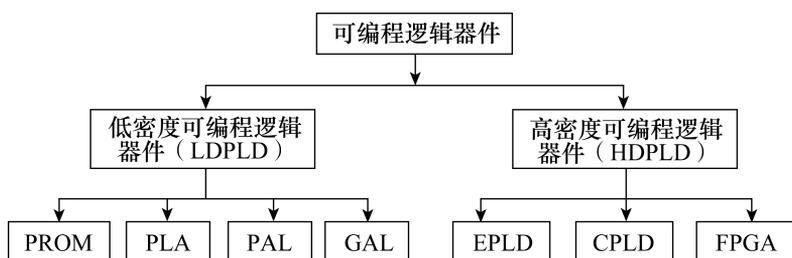


图 2-8 可编程逻辑器件的密度分类

2. 按编程方式分类

可编程逻辑器件按编程方式分为两类：一次性编程（one time programmable, OTP）器件和可多次编程（many time programmable, MTP）器件。OTP 器件属于一次性使用，只允许用户对器件编程一次，编程后不能修改，其优点是可靠性与集成度高，抗干扰性强。MTP 器件可多次重复使用，允许用户对其进行多次编程、修改或设计，特别适用于系统样机的研制和初级设计者。

3. 按元件结构和编程方式分类

根据各种可编程元件的结构及编程方式，PLD 通常又可以分为以下四大类。

(1) 采用一次性编程的熔丝或反熔丝元件的可编程器件，如 PROM、PAL 和 EPLD 等。

如图 2-9 所示，熔丝型开关的编程原理主要是通过熔丝（通常是一段位于两个焊盘之间的金属或多晶硅材料）的熔断来改变逻辑状态，其编程步骤包括供电与使能、设置修调信号、熔丝熔断以及修调完成等。

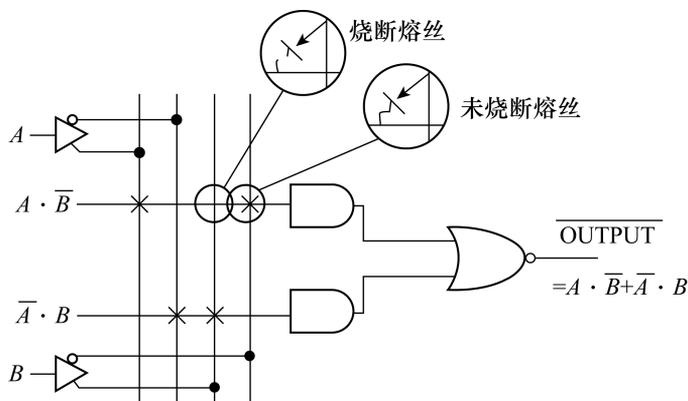


图 2-9 熔丝型开关的编程原理

图 2-10 所示为 PLICE 反熔丝结构，主要层结构分为上层、中间层与下层。上层是多晶硅层，作为导电层，在编程过程中起到关键作用；中间层为 PLICE 介质，属于绝缘部分，它在未编程状态下保持高阻抗，防止电流通过；下层为扩散层，也是一个导电层，可与上层的多晶硅层共同构成反熔丝结构的两端。

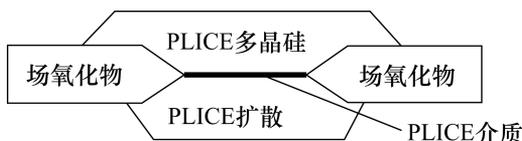


图 2-10 PLICE 反熔丝结构

另外，非晶体反熔丝也是一种特殊的反熔丝结构。如图 2-11 所示，其特点为中间的绝缘层采用非晶硅材料，从而构成了金属-非晶硅-金属的“三明治”结构。具体结构为上下层、导电层和中间绝缘层，其中上下层和导电层均为金属，具有良好的导电性能，而中间绝缘层为非晶硅材料，在编程前呈现绝缘状态，是形成反熔丝特性的关键部分。

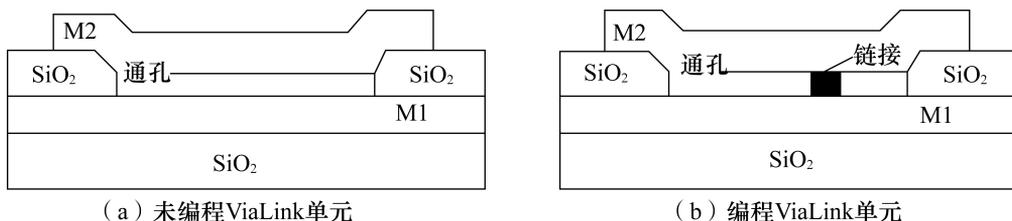


图 2-11 非晶体反熔丝结构

(2) 采用紫外线擦除、电可编程元件，即采用 EPROM、UVC MOS 工艺结构的可多次编程器件，如 EPROM、EEPROM、闪速存储器等。

EPROM 的存储内容不仅可以根据需要来编制，在需要更新存储内容时还可将原存储内容抹去，再写入新的内容。其基本结构是一个浮栅管，相当于一个电子开关，当浮栅中没有注入电子时，浮栅管导通；当浮栅中注入电子后，浮栅管截止。如图 2-12 所示为一种以浮栅雪崩注入型 MOS 管为存储单元的 EPROM，图 2-12 (a) 和图 2-12 (b) 分别为它的基本结构和电路符号，它与普通的 NMOS 管相似，它有 G_1 和 G_2 两个栅极。 G_1 栅没有引出线，被包围在二氧化硅 (SiO_2) 中，称为浮栅； G_2 为控制栅，有引出线。若在漏极和源极之间加上约几十伏的电压脉冲，在沟道中产生足够强的电场，则会造成雪崩，使电子加速跃入浮栅中，从而使浮栅 G_1 带上负电荷。由于浮栅周围都是绝缘的 SiO_2 层，是泄漏电流极，所以一旦电子注入 G_1 栅后，就能长期保存。

当 G_1 栅有电子积累时，该 MOS 管的开启电压变得很高，即使 G_2 栅为高电平，该管仍不能导通，相当于存储了“0”；当 G_1 栅无电子积累时，MOS 管的开启电压较低，当 G_2 栅为高电平时，该管可以导通，相当于存储了“1”。EPROM 器件的上方有一个石英窗口，如图 2-12 (c) 所示。当用光子能量较高的紫外光照射浮栅时， G_1 栅中电子获得了足够的能量，穿过氧化层回到衬底中。如图 2-12 (d) 所示为读出时的电路，它采用了二维译码方式，其中 x_i 、 y_i 为地址译码器的二维输出，CS 为片选信号。如图 2-12 (e) 所示为光抹成全“1”原理图，这样可使浮栅上的电子消失，达到抹去存储信息的目的，相当于存储器又存了全“1”。这种 EPROM 出厂时为全“1”状态，使用者可根据需要写“0”，如图 2-12 (f) 所示为写“0”电路。在写“0”时， x_i 和 y_i 选择线为高电平，P 端加 20 多伏的正脉冲，脉冲宽度为 0.1~1 ms。

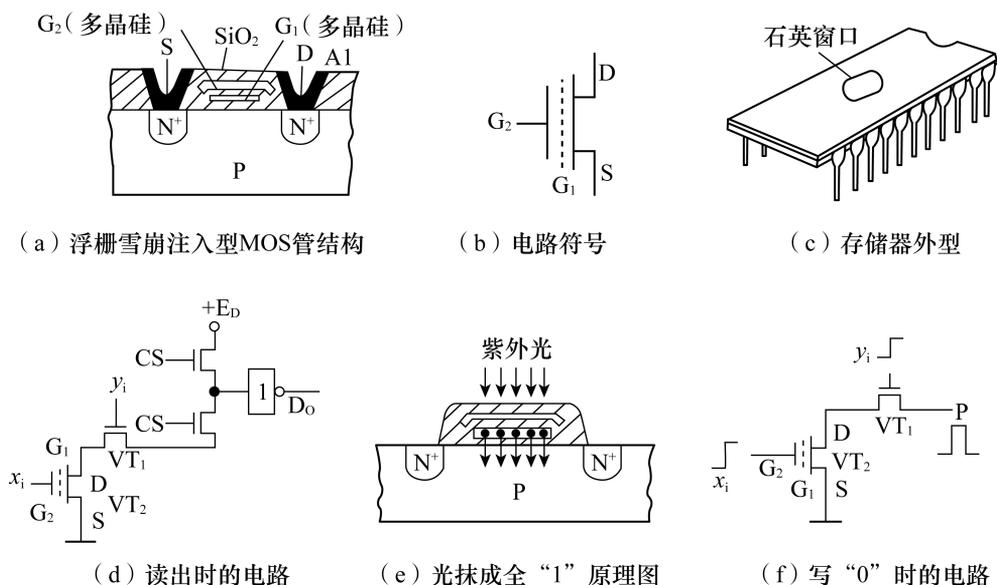


图 2-12 EPROM 存储器

EEPROM 也可写成 E^2 PROM，它是电可擦除的可编程器件。EEPROM 有多种工艺，也基于浮栅技术。如图 2-13 所示为 EEPROM 的存储单元，这是一个具有两个栅极的 NMOS 管，其中 G_1 为控制栅，是一个浮栅，无引出线； G_2 是抹去栅，有引出线。在 G_1 栅和漏极之间有一个小面积的氧化层，其厚度极薄，可产生隧道效应。当 G_2 栅加 20 V 的正脉冲 P_1 时，通过隧道效应，电子由衬底注入 G_1 浮栅，相当于存储了“1”，利用此方法可将存储器抹成全“1”状态。

这种存储器在出厂时，存储内容也为全“1”状态。使用时可根据需要把某些存储单元写“0”。写“0”电路如图 2-13 (d) 所示，此时漏极 D 加 20 V 正脉冲 P_2 ， G_2 栅接地，浮栅上电子通过隧道返回衬底，相当于写“0”。EEPROM 读出时的电路如图 2-13 (e) 所示，这时 G_2 栅加 3 V 的电压，若 G_1 栅有电子积累，则 VT_2 管不能导通，相当于存“1”；若 G_1 栅无电子积累，则 VT_2 管导通，相当于存“0”。

闪存存储器，也称为 Flash Memory，是一种非易失性存储器（non-volatile memory, NVM），其显著特点在于能够在不依赖外部电源供电的情况下，长时间保持信息存储的完整性。闪存存储器通过采用闪存技术，利用电子浮动栅的原理来实现数据的存储和擦除，主要由一系列被称为存储单元的晶体管组成，每个存储单元中包含一个浮动栅，它可以在特定的电压下吸引或排斥电荷。当电荷被吸引到浮动栅时，表示存储单元中的位是“1”；当电荷被排斥时，表示存储单元中的位是“0”。这种电荷状态的变化被用来表示存储的数据，能够在断电后继续保存数据，且支持多次擦写和编程操作，这使得它在各种数据存储需求以及设备间的数据交换中得到了广泛应用。它在设计和工艺上与成熟的 EPROM 的产品十分相似。EPROM 与闪存存储器的区别如图 2-14 所示。

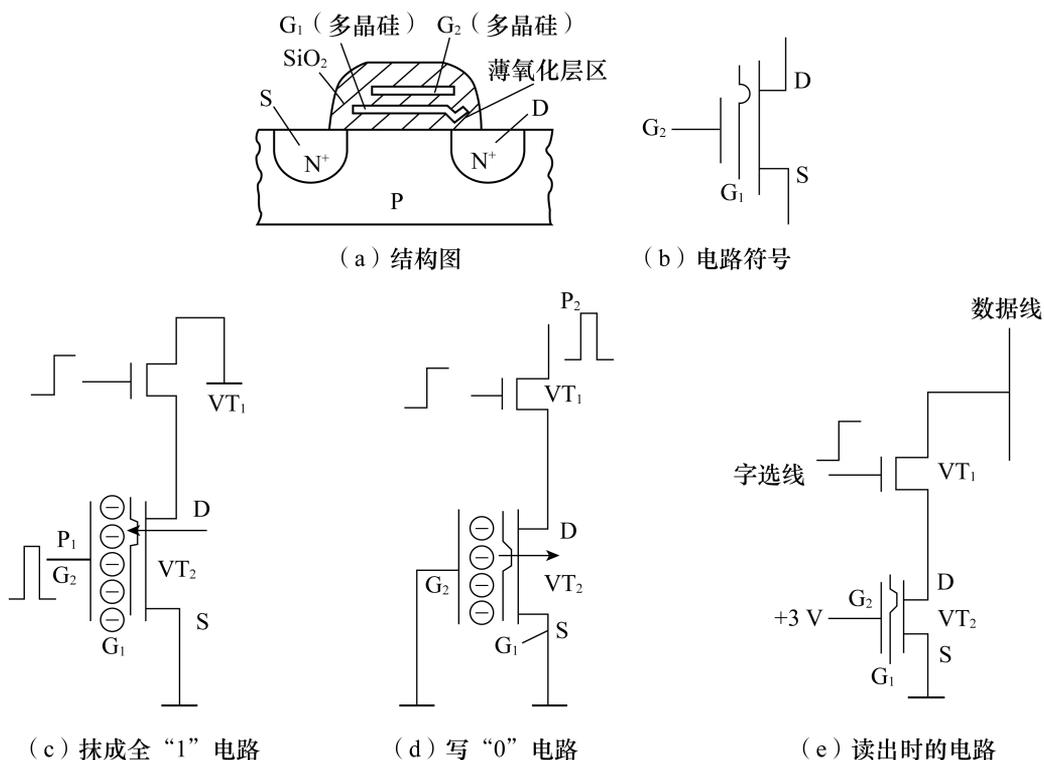


图 2-13 EEPROM 的存储单元

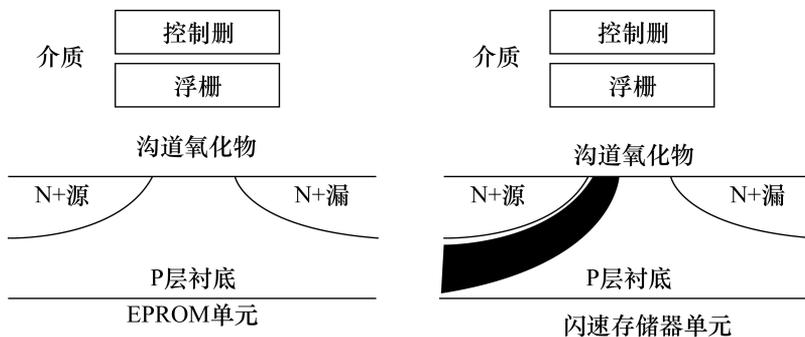


图 2-14 闪速存储器和 EPROM 的区别

(3) 采用电擦除、电可编程元件。其中一种是 EEPROM，另一种是采用静态随机存取存储器 (static random access memory, SRAM)。

SRAM 的基本单元是由五个晶体管组成的存储器。如图 2-15 所示为 SRAM 的基本单元结构。它由两个 CMOS 反相器和一个用来控制读/写的 MOS 传输开关构成，其中每个 CMOS 反相器包含两个晶体管。SRAM 的工作原理是基于交叉耦合反相器的锁存特性。当 SRAM 单元存储数据时，两个 CMOS 反相器相互稳定地保持一个二进制位的状态。当一个节点 (如 SNL) 的电位为高 (表示 1) 时，另一个节点 (如 SNR) 的电位自动为低 (表

示0), 反之亦然。在读写操作时, 通过控制管 (M_5 和 M_6) 的开关状态, 可以允许或阻止数据通过位线传输到存储单元或从存储单元传输出去。SRAM 相较于 DRAM 具有显著的优势, 其存储单元可直接访问, 因此访问速度更快; 在空闲状态下无须刷新, 从而功耗相对较低; 同时, 它还展现出良好的耐用性, 不受辐射等环境因素的影响。然而, SRAM 也存在一些劣势, 如其制造工艺复杂导致的高成本, 存储单元体积较大引起的存储密度较低, 以及断电后数据会丢失的易失性等。

(4) 基于 SRAM 结构的可多次编程器件。目前多数 FPGA 是基于 SRAM 结构的可编程器件。

4. 按结构特点分类

PLD 按结构特点分为阵列型 PLD 和现场可编程门阵列型 FPGA 两大类。阵列型 PLD 的基本结构由与阵列和或阵列组成。简单 PLD (如 PROM、PLA、PAL 和 GAL 等)、EPLD 和 CPLD 都属于阵列型 PLD。现场可编程门阵列型 FPGA 具有门阵列的结构形式, 它由许多可编程单元 (或称逻辑功能块) 排成阵列组成, 称为单元型 PLD。

另外, 随着半导体工艺的线宽不断缩小, 从器件功耗的要求出发, 内芯必须采用低电压。由于 I/O 块与内芯供电电压也可能不同, 这就要求 I/O 块的结构能够兼容多个电压标准, 需要既能接收外部器件的高电压输入信号, 又能驱动任何高电压的器件。1990 年至 2002 年, 工艺线宽与供电电压的关系如图 2-16 所示, 一般来说, 工艺线宽越小, 对功耗的要求就越高, 工作电压就必须降低。如工艺线宽为 $1.2 \sim 0.5 \mu\text{m}$ 时, 器件一般采用 5 V 电压供电; 当工艺线宽为 $0.35 \mu\text{m}$ 时, 器件的供电电压为 3.3 V, 此时 I/O 块与内芯的供电电压相同; 当工艺线宽为 $0.25 \mu\text{m}$ 时, I/O 块与芯片内芯的供电电压不再相同, 内芯的供电电压为 2.5 V, I/O 块的供电电压为 3.3 V, 并且能兼容 5 V 和 3.3 V 的器件; 当工艺线宽为 90 nm 时, 器件应采用 1 V 的供电电压, 而 I/O 要能够兼容 1.3 V 和 1.8 V 的器件。

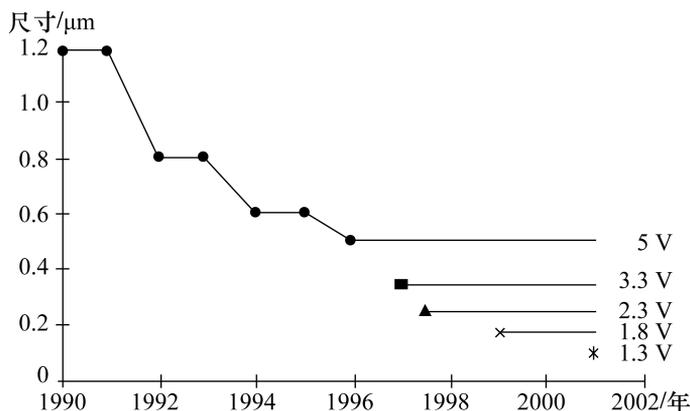


图 2-16 工艺线宽与供电电压的关系

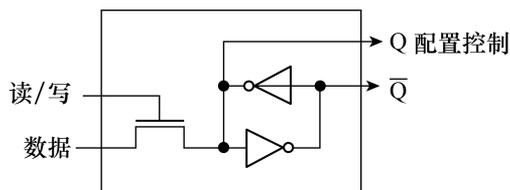


图 2-15 SRAM 的基本单元结构

2.2 CPLD 的结构与工作原理

2.2.1 宏单元结构

宏单元结构通常包含一个查找表、一个可配置的触发器（*D* 型或 *JK* 型）和相关的控制逻辑，它们可以执行多种不同的逻辑功能。

每个宏单元主要由逻辑阵列、乘积项选择矩阵和可编程触发器三个功能块组成。MAX7000S 系列的单个宏单元的结构如图 2-17 所示。逻辑阵列实现组合逻辑功能，它可给每个宏单元提供五个乘积项。“乘积项选择矩阵”分配这些乘积项作为或门和异或门的主要逻辑输入，以实现组合逻辑函数；每个宏单元的一个乘积项可以反相后回送到逻辑阵列。这个可共享的乘积项能够连到同一个 LAB 中任何其他乘积项上。

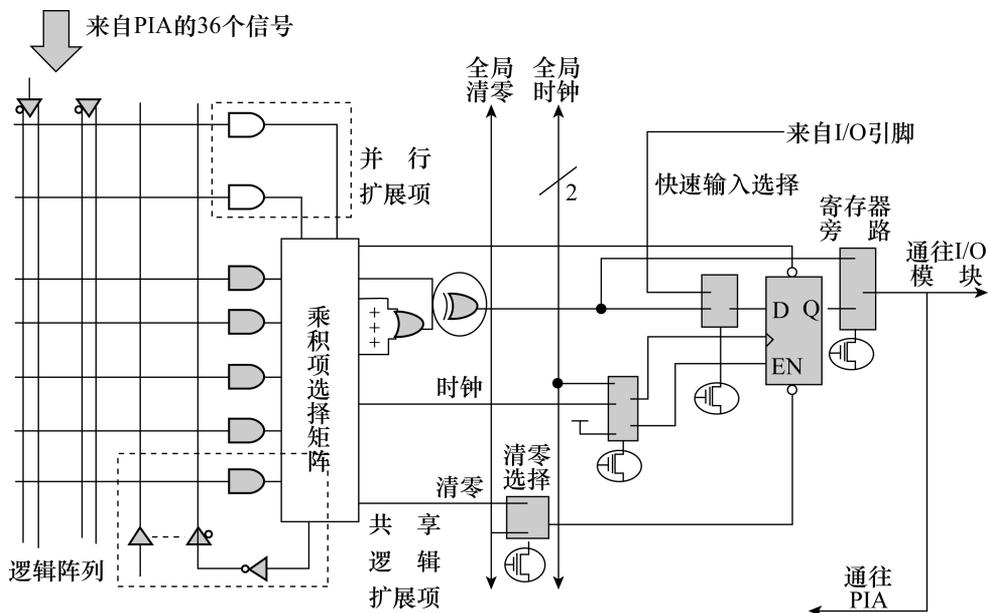


图 2-17 MAX7000S 系列的单个宏单元结构

2.2.2 CPLD 器件结构

CPLD 是一种集成电路，它包含了多个可编程的逻辑块和可编程的互连资源，允许实现各种数字逻辑功能。其基本结构包括逻辑阵列、扩展乘积项、可编程连线阵列和输入/输出单元四个部分。

(1) 逻辑阵列。这是 CPLD 的核心，由许多小的逻辑单元组成，这些逻辑单元可以是宏单元或 LAB。每个宏单元通常包含一个或多个 LUT、触发器和一些辅助电路。逻辑阵列

块及它们之间的连线构成 MAX7000 器件的结构，如图 2-18 所示。每个 LAB 由 16 个宏单元组成，多个 LAB 通过可编程连线阵列（programmable interconnect array, PIA）和全局总线连接。

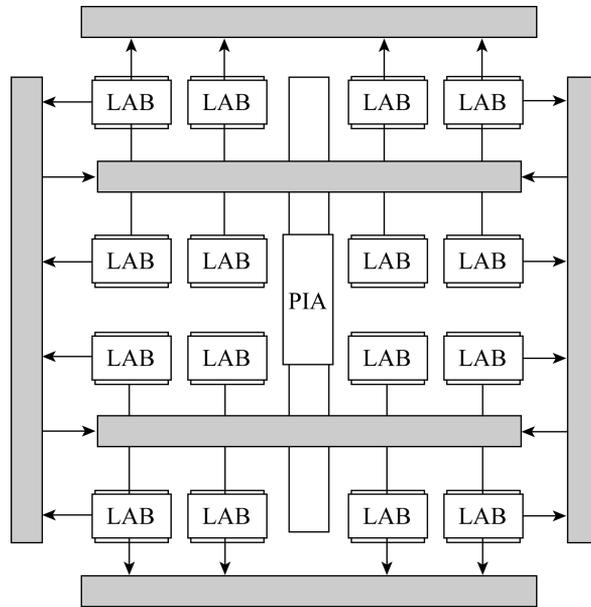


图 2-18 MAX7000 器件的结构

(2) 扩展乘积项。CPLD 的扩展乘积项（product terms）是其逻辑阵列的一个关键特性，用于实现数字逻辑功能。在 CPLD 中，乘积项通常指的是与阵列中的每个乘积项，它们可以由多个输入信号通过逻辑与操作组合而成。扩展乘积项有共享扩展项和并联扩展项两种。

共享扩展项就是由每个宏单元提供一个未使用的乘积项，并将它们反相反馈到逻辑阵列，便于集中使用。每个 LAB 有 16 个共享扩展项，可被 LAB 中的任何宏单元使用和共享，以实现复杂的逻辑函数。采用共享扩展项会增加一个短的延时。如图 2-19 所示为共享扩展项结构。

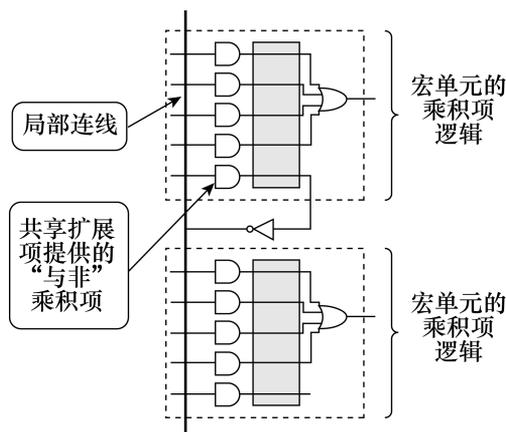


图 2-19 共享扩展项结构

并联扩展项是一些宏单元中没有使用的乘积项，并且这些乘积项可分配到邻近的宏单元去实现快速复杂的逻辑函数。并联扩展项允许多达 20 个乘积项直接馈送到宏单元的或逻辑，其中 5 个乘积项是宏单元本身提供的，15 个并联扩展项是由 LAB 中邻近宏单元提供的，其结构如图 2-20 所示。

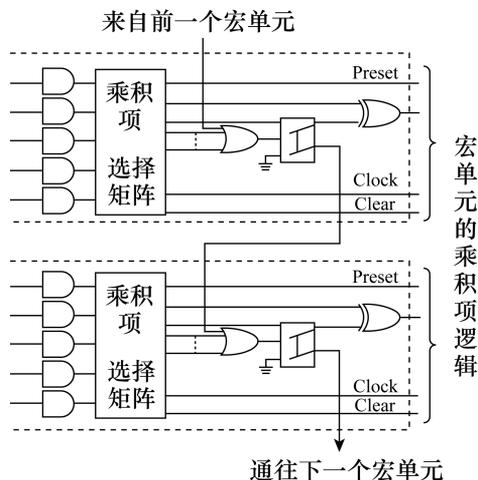


图 2-20 并联扩展项结构

(3) 可编程连线阵列 (PIA)。PIA 信号布线到 LAB 的方式是 CPLD 内部架构的关键部分。具体来说，PIA 负责将信号从器件的 I/O 引脚或其他内部资源连接到 LAB，以及实现 LAB 之间的互连。图 2-21 是 PIA 信号布线到 LAB 的方式。

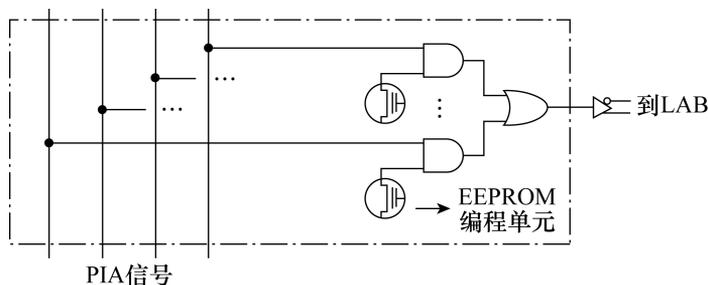


图 2-21 PIA 信号布线到 LAB 的方式

(4) 输入/输出单元 (I/O)。这些单元提供了与外部器件通信的接口。每个 I/O 单元都可以配置为不同的电平标准，如 TTL、CMOS、LVCMOS 以及 LVTTTL 等。如图 2-22 所示为 EPM7128S 器件的 I/O 控制块。

CPLD 的设计灵活性、可重配置性和中等规模等优点使其在多种应用中都有广泛的使用，包括数字信号处理、工业控制、通信和消费电子。与 FPGA 相比，CPLD 通常能提供更低的功耗和成本，以及更简单的编程模型。

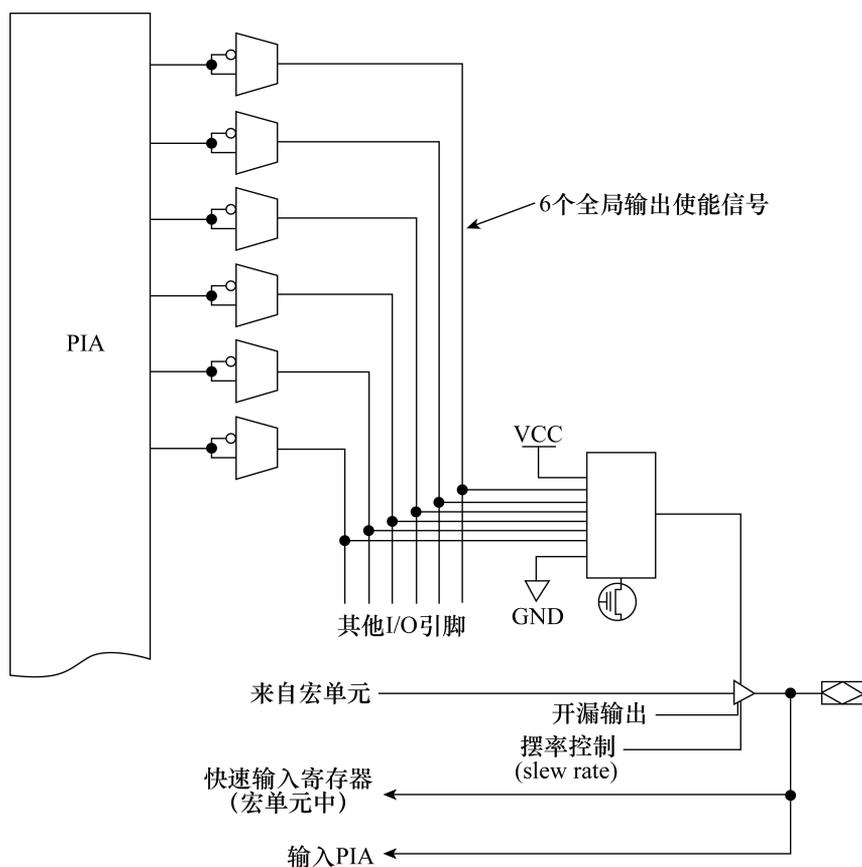


图 2-22 EPM7128S 器件的 I/O 控制块

2.2.3 CPLD 产品简介

CPLD 的应用已深入网络、仪器仪表、汽车电子、数控机床、航天测控设备等多个领域。其优点包括编程灵活、集成度高、设计开发周期短、适用范围宽、开发工具先进、设计制造成本低、对设计者的硬件经验要求低、标准产品无须测试、保密性强，以及价格适中。这些优点使得 CPLD 被广泛应用于产品的原型设计和产品生产中，尤其是中小规模通用数字集成电路的应用场合。

一个典型的 CPLD 设计流程包括原理图绘制、硬件描述语言编写、编译、仿真、管脚输入输出锁定和代码生成，最后通过下载电缆将代码传送到 CPLD 芯片。

目前，市场上有多种 CPLD 产品，主要由 Altera、Lattice、Xilinx 等公司生产。一些常见的 CPLD 产品包括 Altera 公司生产的 EPM7128S、Lattice 公司生产的 LC4128V 和 Xilinx 公司生产的 XC95108 等。

2.3 FPGA 的结构与工作原理

2.3.1 FPGA 结构

FPGA 是一种高度灵活的半导体器件，它包含大量的 CLB、可编程的互连资源以及可编程的 I/O 块。FPGA 能够根据设计者的需求实现各种数字电路功能。

如图 2-23 所示，FPGA 的基本结构包括以下六个部分。

(1) 可配置逻辑块 (CLB)：这是 FPGA 的核心，通常包含查找表、触发器、多路复用器和其他辅助逻辑。

(2) 可编程 I/O 块：这些块提供了与外部器件通信的接口。每个 I/O 块都可以配置为不同的电平标准，如 TTL、CMOS、LVCMOS、LVTTTL 等，并且可以支持差分信号。

(3) 时钟管理：FPGA 通常包含时钟管理电路，如锁相环 (PLL) 或延迟锁定环 (DLL)，用于生成和调整内部时钟信号。

(4) 块 RAM：许多 FPGA 还包含块 RAM，这是一种可配置的存储资源，可以用于实现缓存、先进先出 (FIFO) 存储器或简单的数据存储功能。

(5) 数字信号处理 (DSP) 块：一些 FPGA 型号还包含专用的 DSP 块，用于加速数字信号处理任务，如滤波、FFT 等。

(6) 配置存储器：FPGA 在掉电后能够保持其配置，这是因为它们包含非易失性配置存储器，如 EEPROM 或 Flash。上电时，这些存储器中的数据会被加载到 FPGA 的逻辑阵列中，以确定其功能。

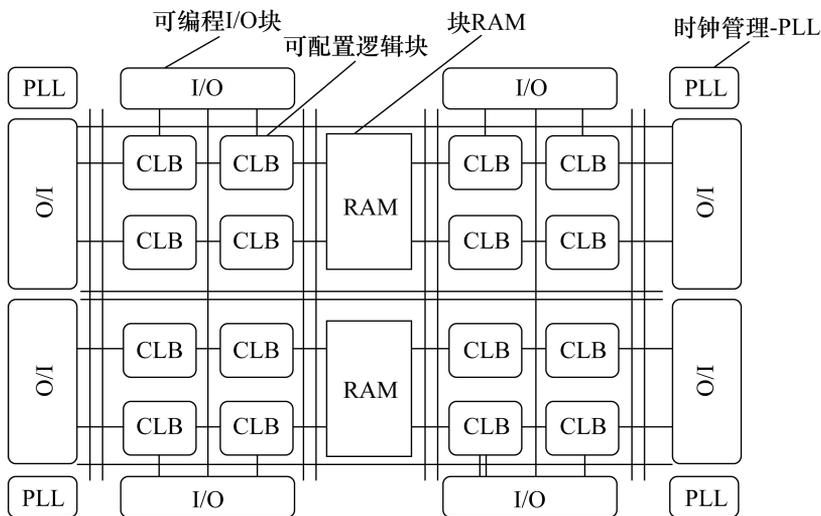


图 2-23 FPGA 的基本结构

FPGA 的设计灵活性、可重配置性和大规模特性使其在多种应用中都有广泛的使用,包括数字信号处理、工业控制、通信、医疗设备和航空航天等。与 CPLD 相比, FPGA 通常可提供更高的逻辑密度和更复杂的逻辑功能,但功耗和成本也相对较高。

2.3.2 可编程逻辑模块

FPGA 的可编程逻辑模块 (programmable logic block, PLB) 是其核心组成部分,它们为设计者提供了实现各种数字逻辑功能的能力。这些模块通常被称为可配置逻辑块或逻辑单元,具体名称和结构可能因不同的 FPGA 厂商和型号而异。FPGA 可编程逻辑模块的这种结构使得设计者能够通过编程来配置和修改逻辑功能,从而实现高度灵活和可定制的数字电路设计。

2.3.3 查找表电路结构

FPGA 中的查找表 (look-up table, LUT) 是实现组合逻辑功能的基本单元。LUT 本质上是一个小的存储单元,它可以根据输入信号的组合来提供相应的输出。其设计允许用户通过编程来定义输入和输出之间的关系。如图 2-24 所示为 LUT 单元。

LUT 单元的内部结构如图 2-25 所示。一个 n 输入的 LUT 包含 2^n 个存储单元,每个存储单元可以存储一个比特的信息 (0 或 1)。这些存储单元的输出连接到一个多路复用器的输入端,多路复用器的输出就是 LUT 的输出。输入信号用于选择多路复用器中的特定路径,从而决定哪个存储单元的值被输出。

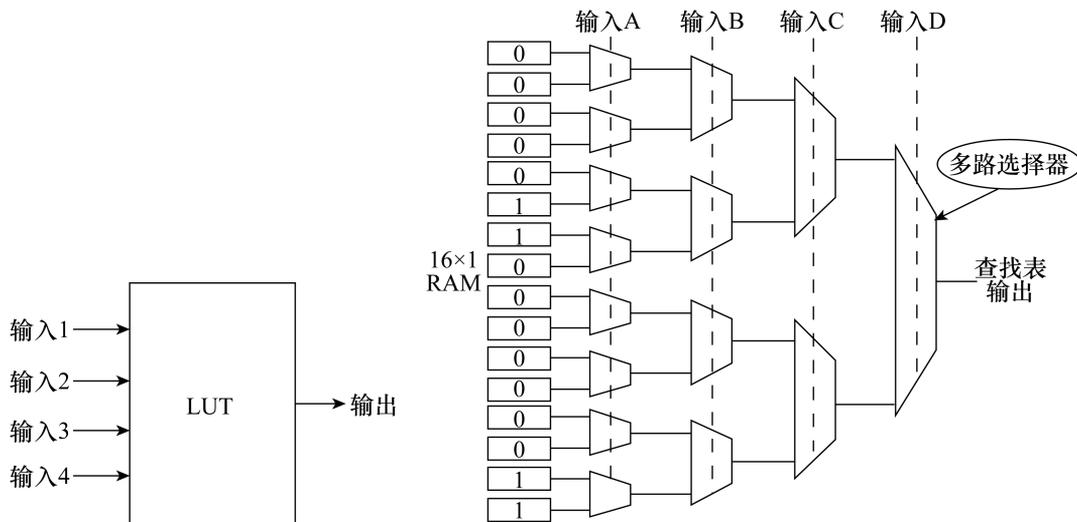


图 2-24 LUT 单元

图 2-25 LUT 单元的内部结构

2.3.4 可编程 I/O 模块

FPGA 的可编程 I/O 模块是 FPGA 与外部世界通信的接口,它提供了高度的灵活性和可配置性,允许设计者根据特定的应用需求来调整 I/O 端口的行为。其主要特点和功能包括:

(1) 电平转换：I/O 模块可以配置不同的电平标准，如 TTL、CMOS、LVCMOS、LVTTL、PCI、LVDS 等，以适应不同的电压要求和信号标准。

(2) 驱动能力：可调整 I/O 模块的输出驱动能力，以匹配负载的需求。例如，可以将驱动能力配置为强驱动或弱驱动。

(3) 输入/输出寄存器：I/O 模块通常包含寄存器，用于缓冲输入信号或驱动输出信号。这些寄存器可被设置为同步或异步工作模式。

(4) 三态控制：许多 FPGA 的 I/O 模块支持三态功能，允许设计者将输出置为高阻态，这在多路复用或总线应用中非常有用。

2.3.5 FPGA 专用元件

FPGA 专用元件集成了高度可编程逻辑元件的器件，其核心结构由可编程逻辑块和查找表等组成，并配备了可编程输入/输出单元以实现与外部世界的通信。此外，FPGA 还包含嵌入式块 RAM 用于数据存储，以及 DSP 计算单元提供算力支持。为了增强功能性和支持特定任务，FPGA 还集成了专用硬核模块，如锁相环、硬核处理器以及串行/解串行接口等。这些专用元件共同赋予了 FPGA 高度的灵活性和可编程性，使其在现代电子系统设计中成为不可或缺的重要组成部分。

(1) FPGA 配置电路。其配置电路原理如图 2-26 所示，主要通过一个串行接口实现，该接口包含四个关键引脚，即 DATA、DCLK、nCs 和 ASDI。标准中更常见的是 nSTATUS 或特定于配置的信号，但在此假设其为配置流程中的一部分控制信号，用于简化说明。系统启动时，FPGA 与串行配置器件均经历一个上电复位阶段，这期间 FPGA 将 nSTATUS 和 CONF_DONE 信号置为低电平，以此指示当前尚未完成配置。此复位周期大约持续 100 ms，随后 FPGA 释放 nSTATUS 信号并转入配置模式。

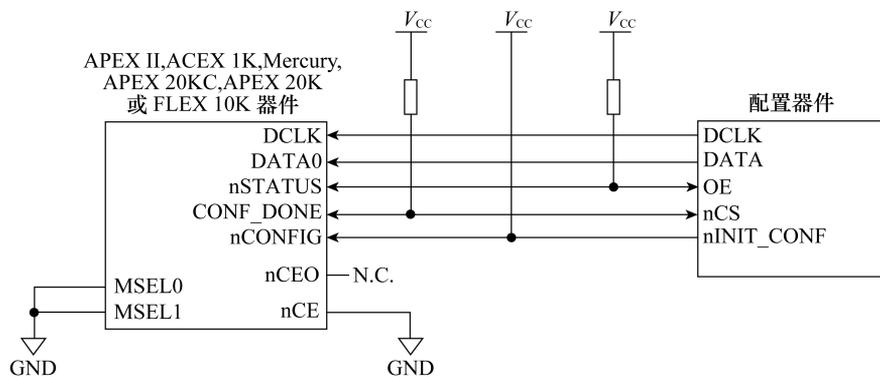


图 2-26 FPGA 配置电路原理

在配置模式下，FPGA 通过拉低 nCS 引脚来激活串行配置器件，同时利用其内部振荡器生成 DCLK 信号，该信号频率范围为 14~20 MHz，典型值为 17 MHz，为数据和控制信号的传输提供精准的时钟基准。ASDI 引脚在 DCLK 的下降沿发送控制指令，而 DATA 引脚则负责在 DCLK 的每个上升沿串行发送配置数据。串行配置器件则遵循相反的边沿逻辑：在 DCLK 的上升沿捕获输入信号，并在其下降沿输出配置数据给 FPGA。FPGA 则在

DCLK 的下降沿发送控制信号，并在随后的上升沿捕获配置数据。一旦配置过程顺利完成，FPGA 会释放 CONF_DONE 信号，此信号随后由外部电路拉至高电平，标志着 FPGA 已准备好进行初始化及后续操作。

(2) EPCS 器件配置 FPGA 电路。如图 2-27 所示，在被动串行 (passive serial, PS) 配置模式下，配置过程由 EPC 配置器件或外部计算机 (及微处理器) 主导。此模式下，配置数据在 DCLK 的每个精确上升沿，通过 DATA0 引脚以串行方式被送入 Cyclone 器件。EPC1、EPC2 和 EPC1441 器件将配置数据存放于 EPROM 中，并按照内部晶振产生的时钟频率将数据输出；OE、nCS 和 DCLK 引脚提供了地址计数器和三态输出缓存的控制信号；配置器件将配置数据按串行的比特流由 DATA 引脚输出。

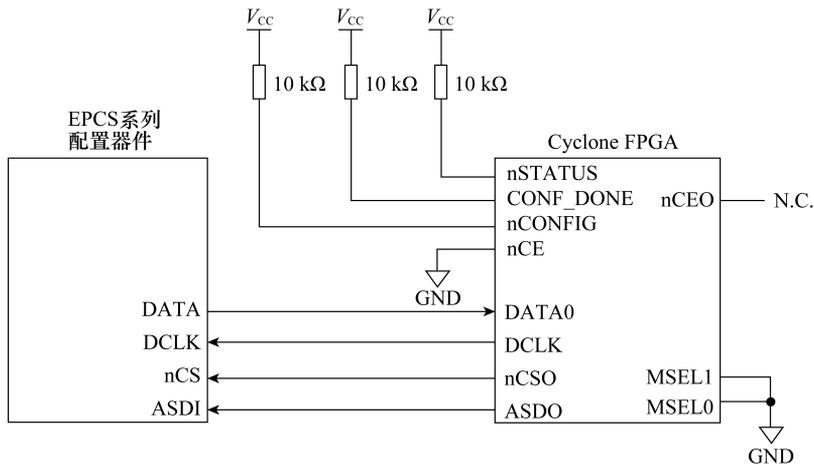


图 2-27 EPCS 器件配置 FPGA 电路

(3) 89C52 单片机配置 FPGA 的电路原理。图 2-28 中的单片机采用常见的 89C52，FLEX 10K 的配置模式选为 PS 模式。由于 89C52 的程序存储器是内嵌于芯片的 Flash ROM，设计的保密性较好。另外，若把图中的“其他功能模块”换成无线接收模块，可实现系统的无线升级，因此该电路有很大的扩展余地。

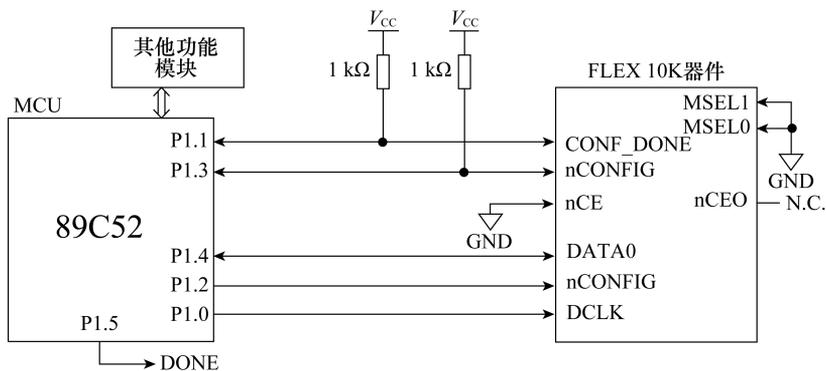


图 2-28 89C52 单片机配置 FPGA 的电路原理

(4) 使用 PC 并行口配置 FPGA。基于 SRAM LUT 结构的 FPGA 不属于 ISP 器件，它是在线可重配置方式 (in circuit reconfigurability, ICR) 改变芯片内部的结构来进行硬件验证

的。利用 FPGA 进行电路设计时，可通过下载电缆与 PC 机的并口连接，将设计文件编程下载到 FPGA。使用 PC 机的并口，通过 ByteBlaster 下载电缆对多个 FPGA 器件进行配置。

图 2-29 是 PS 模式下 FLEX 10K 的配置时序。图中标出了 FPGA 器件的三种工作状态：配置状态、初始化状态与用户模式（正常工作状态）。配置状态是指 FPGA 正在配置的状态，用户 I/O 全部处于高阻态；初始化状态指配置已经完成，但 FPGA 器件内部资源（如寄存器）还未复位完成，逻辑电路还未进入正常状态；用户模式是指 FPGA 器件已得到配置，并处于正常工作状态，用户 I/O 在正常工作。

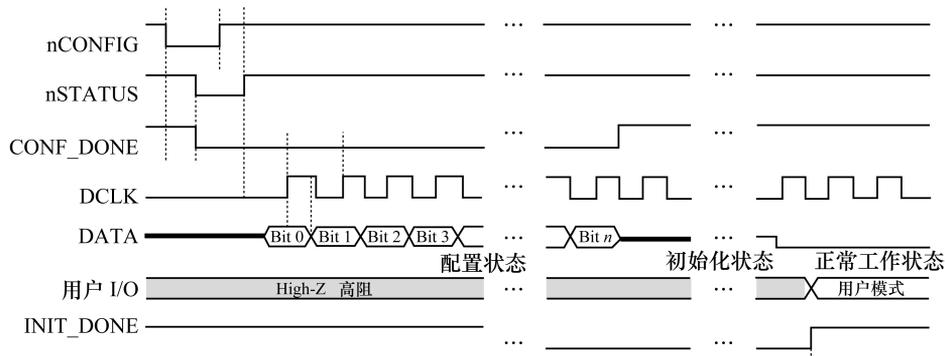


图 2-29 PS 模式下 FLEX 10K 配置时序

当设计的数字系统比较大，需要不止一个 FPGA 器件时，为每个 FPGA 器件都设置一个下载口显然是不实用的，而 Altera 器件的 PS 模式支持多个器件进行配置。对于 PC 机而言，除了在软件上要设置支持多器件外，再加一条 ByteBlaster 下载电缆即可对多个 FPGA 器件进行配置。图 2-30 的电路给出了多 FPGA 芯片配置电路。

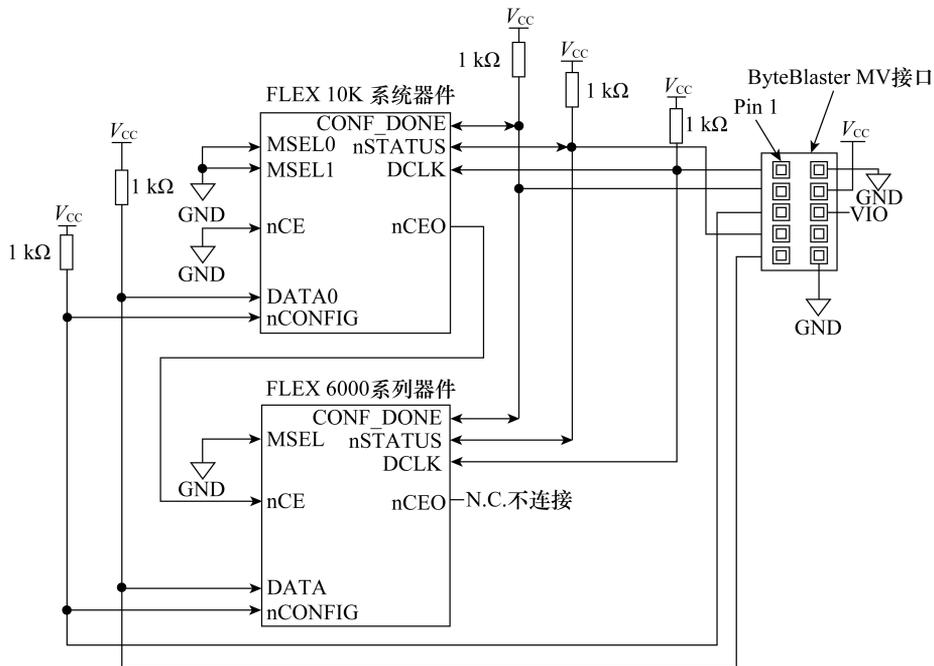


图 2-30 多 FPGA 芯片配置电路

2.4 CPLD 与 FPGA 的发展趋势

CPLD 和 FPGA 的发展趋势主要体现以下四个方面。

(1) 行业需求增长。CPLD 和 FPGA 在各行业中的应用需求持续增长，尤其是在移动通信、远程通信基础设施、消费电子等领域。这得益于 CPLD 和 FPGA 的灵活性和高性能，使它们能够满足不同应用场景的需求。

(2) 技术创新。随着技术的发展，CPLD 和 FPGA 正变得更加高效和强大。例如，FPGA 正在通过软件工具和预构建的 IP 模块简化 AI 模型的集成和优化，使得更多的开发者能够将 AI 特性集成到设备中。

(3) 应用领域拓展。CPLD 和 FPGA 的应用领域正在不断拓展，从传统的通信和计算领域扩展到汽车、医疗、消费电子、工业等多个领域。这得益于 CPLD 和 FPGA 的灵活性和可定制性，使它们能够适应不同行业的特定需求。

(4) 市场竞争加剧。随着需求的增加，CPLD 和 FPGA 的市场竞争也在加剧。各大厂商正在通过技术创新和产品优化来争夺市场份额，这将进一步推动 CPLD 和 FPGA 技术的发展。

PLD 显著的优势在于它能极大地缩短产品的研制周期，通过灵活配置逻辑功能来快速响应市场变化，从而有效降低了设计成本。同时，其高度的设计灵活性使得工程师能够轻松调整和优化设计方案，以适应不同的应用需求，进一步提升了开发效率和项目的市场竞争力。

因此，PLD 的发展趋势展现出多元化的路径，它们正逐步向高密度和大规模方向发展，以满足日益复杂的应用需求。同时，向系统内可重构性的演进，赋予了 PLD 在运行时动态调整逻辑功能的能力，增强了系统的灵活性和适应性。此外，低电压、低功耗的设计趋势符合绿色节能的理念，有助于提高设备的能效比。随着技术的进步，高速可预测延时器件的发展也将成为重点，以满足对实时性和性能稳定性要求极高的应用场景。最后，混合可编程技术融合了不同技术的优势，预示着未来 PLD 将更加智能、综合和强大。

2.5 PLD 测试技术

PLD 测试技术是针对 PLD 器件进行功能验证和性能评估的一系列技术手段，主要目的是确保 PLD 器件在设计和制造过程中可实现预定的功能和达到性能要求。通过测试，可以检测器件是否存在缺陷，验证其逻辑功能是否正确，以及评估其性能是否达标等。

ISP 技术是 20 世纪 80 年代末 Lattice 公司首先提出的一种先进的编程技术。在系统可编程是指对器件、电路板或整个电子系统的逻辑功能可随时进行修改或重构的能力。支持 ISP 技术的 PLD 称为在系统可编程器件 (ISP-PLD)，例如 Lattice 公司生产的 ispLSI1000 ~ ispLSI8000 系列器件就属于 ISP-PLD。

ISP 方式是当系统上电并正常工作时，计算机就可以通过 CPLD 器件拥有的 ISP 接口直接对其进行编程，器件被编程后立即进入正常工作状态。CPLD 的编程和 FPGA 的配置可以使用专用的编程设备，也可以使用下载电缆。例如用 Altera 公司的 ByteBlaster 并行下载电缆，将 PC 机的并行打印口与需要编程或配置的器件连接起来，在 Quartus II 工具软件的控制下就可以对 Altera 公司的多种 CPLD、FPGA 进行编程或配置。如图 2-31 所示为 CPLD 编程下载连线框架。

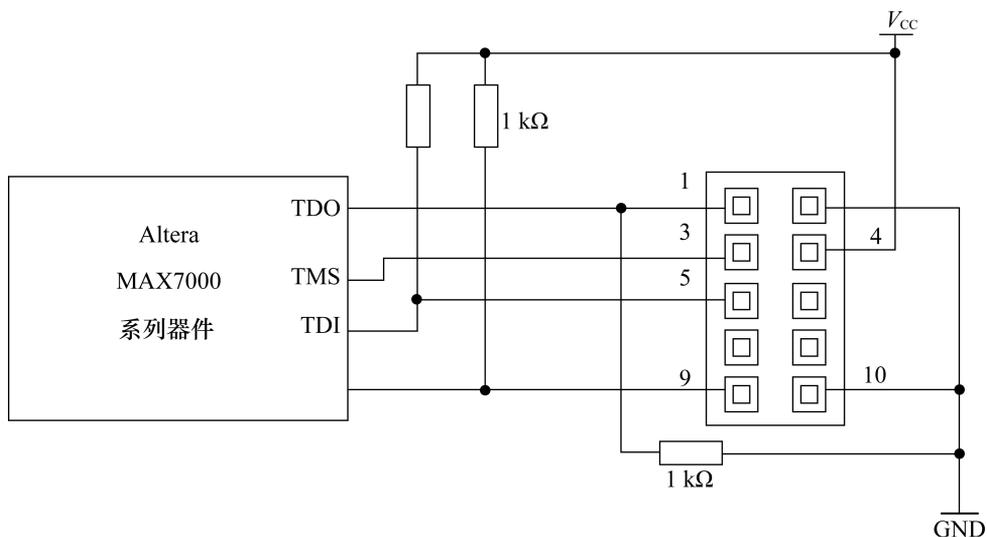


图 2-31 CPLD 编程下载连线框架

另外，边界扫描测试 (boundary-scan testing, BST) 是针对器件密度及 I/O 接口数增加，信号注入和测取难度越来越大的情况而提出的一种新的测试技术。它是由联合测试活动组织 JTAG 提出来的，而后 IEEE 对此制定了测试标准，称为 IEEE 1149.1 标准。BST 主要解决芯片的测试问题。其中的 JTAG 接口主要用作 BST，这样可以省去专用的编程接口，减少系统的引出线。采用 JTAG 模式对 CPLD 编程下载的连线如图 2-32 所示。这种连线方式既可以对 CPLD 进行测试，也可进行编程下载。

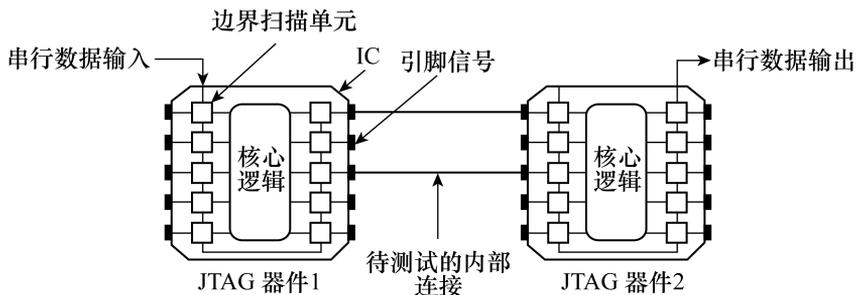


图 2-32 采用 JTAG 模式对 CPLD 编程下载的连线