

8.4 可编程逻辑器件

1. 概述

自20世纪60年代以来，数字集成电路已经历了从SSI、MSI、LSI到VLSI的发展过程。数字集成电路按照芯片设计方法的不同大致可以分为三类：① 通用型中、小规模集成电路；② 用软件组态的大规模、超大规模集成电路，如微处理器、单片机等；③ 专用集成电路(ASIC-Application Specific Integrated Circuit)。

ASIC是一种专门为某一应用领域或为专门用户需要而设计、制造的LSI或VLSI电路，它可以将某些专用电路或电子系统设计在一个芯片上，构成单片集成系统。

可编程逻辑器件 (PLD, Programmable Logic Device) 就是一种ASIC, 它是厂家作为一种通用器件生产的半定制电路, 用户可以利用软、硬件开发工具对器件进行设计和编程, 使之实现所需要的逻辑功能。

可编程逻辑器件按集成度分有低密度可编程逻辑器件 (LDPLD) 和高密度可编程逻辑器件 (HDPLD) 两类。

2. 低密度可编程逻辑器件

低密度可编程逻辑器件的集成密度约为每片 700 个等效门以下, 它主要包括PROM、FPLA、PAL和GAL四种器件。

b. PLD的基本结构

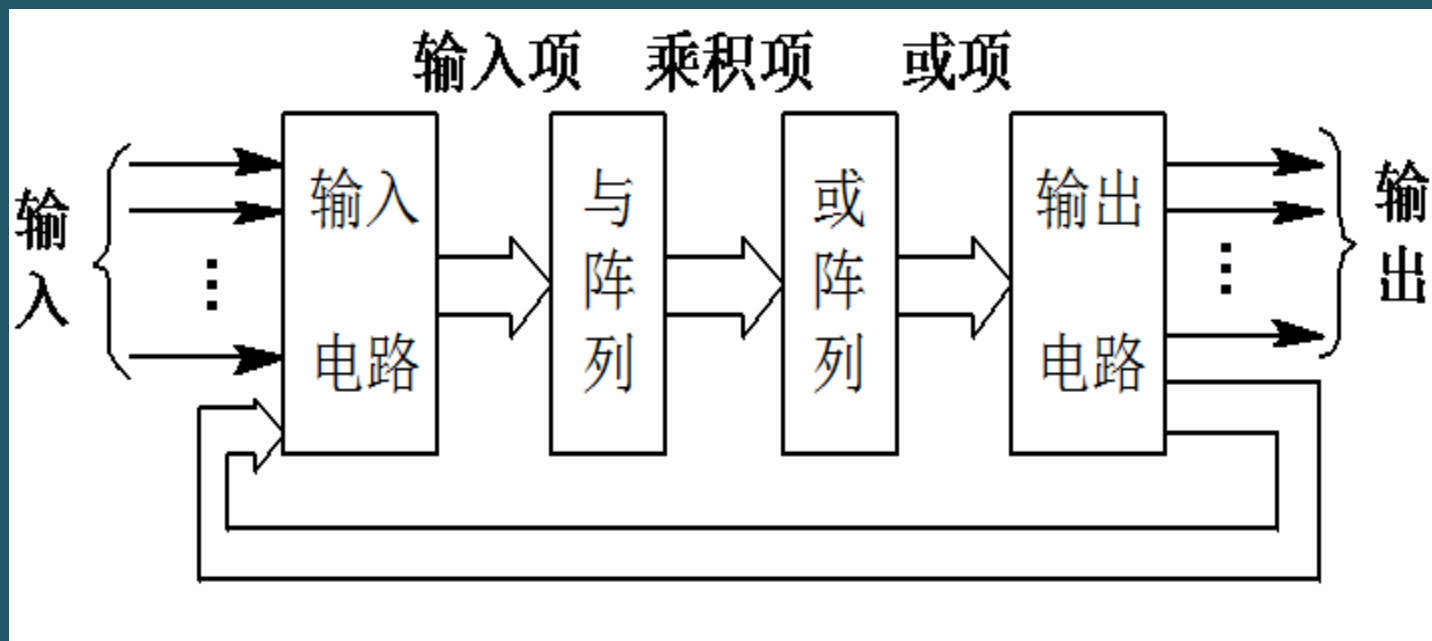


图 8-16 PLD基本结构框图

② 现场可编程逻辑阵列 (FPLA)

1970年制成的PROM是最早出现的PLD。由前面分析可知，PROM由全译码的与阵列和可编程的或阵列组成，由于其阵列规模大，速度低，因而它的基本用途是用作存储器，如软件固化、显示查寻等。

FPLA是20世纪70年代中期在PROM基础上发展起来的PLD，它的与阵列和或阵列均可编程，比PROM使用起来更加灵活。

③ 通用阵列逻辑(GAL)

GAL是Lattice公司于1985年首先推出的新型可编程逻辑器件。它采用了电擦除、电可编程的E²CMOS工艺制作，可以用电信号擦除并反复编程上百次。GAL器件的输出端设置了可编程的输出逻辑宏单元(OLMC-Output Logic Macro Cell)，通过编程可以将OLMC设置成不同的输出方式。这样GAL器件可以实现PAL器件所有的各种输出电路工作模式，因此称为通用可编程逻辑器件。

GAL器件分为两大类：一类为与PAL相似的普通型GAL，如GAL16V8、ispGAL16Z8、GAL20V8；另一类为新型GAL，其与或阵列均可编程，与FPLA结构相似，主要有GAL39V8。

8.6 高密度可编程逻辑器件

1. 概述

- 高密度可编程逻辑器件 (HDPLD)主要包括EPLD、CPLD和FPGA三种类型。
- EPLD是20世纪80年代中期由Altera公司推出的可擦除可编程逻辑器件。它采用了CMOS和UVEPROM工艺制作，和GAL结构类似，但比GAL器件的集成度高很多。
- CPLD是指复杂可编程逻辑器件，它是在EPLD基础上发展起来的器件。CPLD采用E²CMOS工艺制作，有些内部

还集成了RAM、FIFO等存储器，因此，CPLD比EPLD功能更强，使用更灵活，因而得到了广泛应用。

- FPGA是20世纪80年代中期由Xilinx公司推出的一种新型可编程逻辑器件，其电路结构和编程方式与CPLD完全不同。CPLD的主体是与或阵列，并以可编程逻辑单元为基础，可编程连线集成在一个全局布线区，因此称为阵列型HDPLD；FPGA则是以基本门单元为基础，构成门单元阵列，可编程连线分布在门单元之间的布线区，因此称为单元型HDPLD。FPGA采用CMOS-SRAM工艺制作，因此断电后数据随之丢失，在工作前需要从芯片外的EPROM中加载配置数据。

2. 现场可编程门阵列FPGA

FPGA采用了逻辑单元阵列 (Logic Cell Array, LCA) 的概念, 内部包括可配置逻辑块 (CLB-Configurable Logic Block)、输入/输出模块 (Input Output Block, IOB) 和互连资源 (Interconnect Resource, IR) 三个部分, 图8-24所示为FPGA的基本结构图。与传统逻辑电路和门阵列 (如PAL, GAL及CPLD器件) 相比, FPGA利用小型查找表 (如 16×1 的RAM) 来实现组合逻辑, 每个查找表连接到一个D触发器的输入端, 触发器再来驱动其它逻辑电路或驱动I/O, 由此构成了既可实现组合逻辑功能又可实现时序逻辑功能的基本单元模块, 这些模块利用

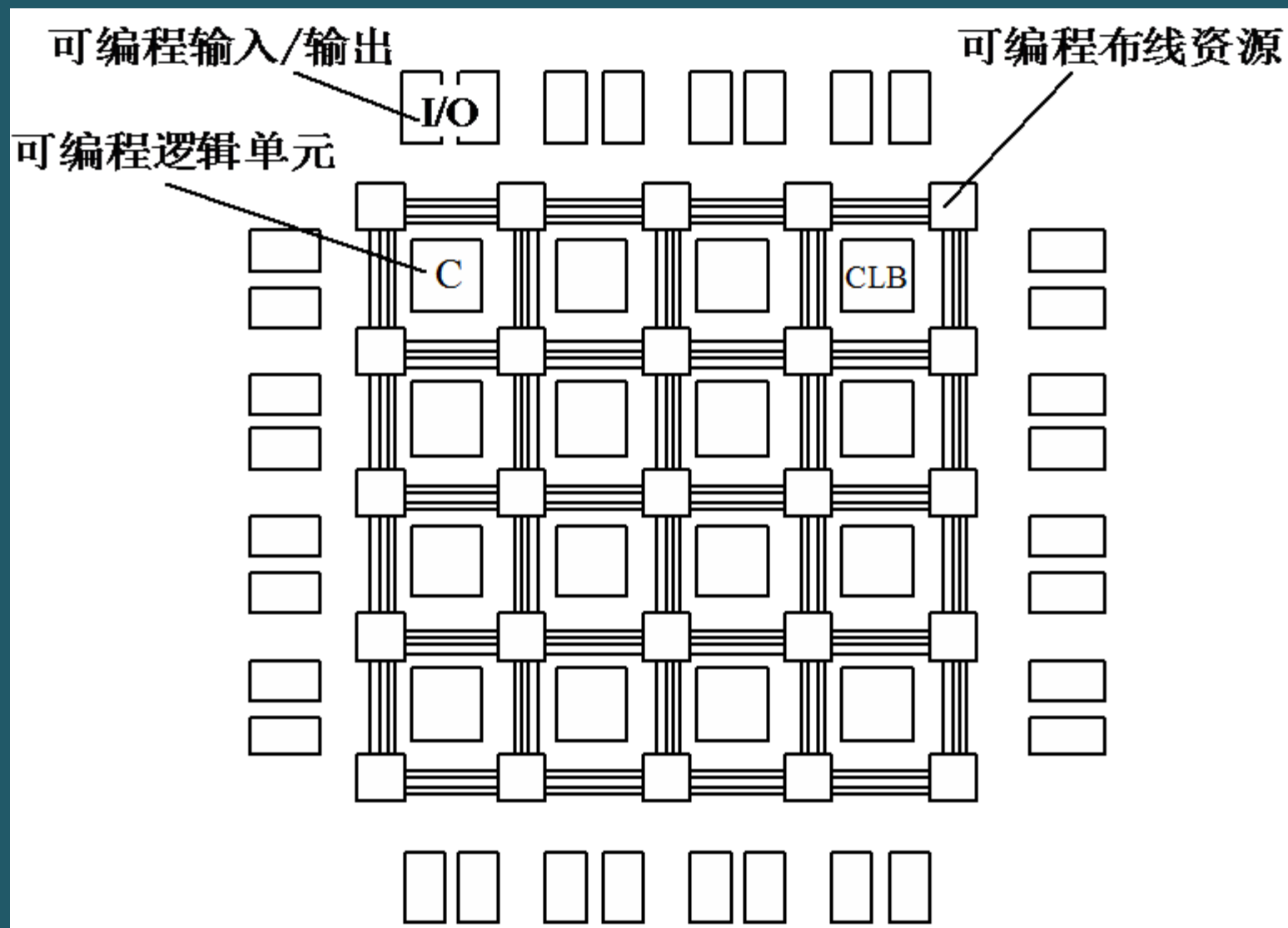


图 8-24 FPGA的基本结构

金属连线互相连接或连接到I/O模块。FPGA的逻辑功能是通过向内部静态存储单元加载编程数据来实现的，并且允许无限次的编程。

可配置逻辑块CLB是实现用户功能的基本单元，它们通常规则地排列成一个阵列，散布于整个芯片；可编程输入/输出模块(IOB)主要完成芯片上逻辑与外部封装脚的接口，它通常排列在芯片的四周；可编程互连资源(IR)包括各种长度的连线线段和一些可编程连接开关，它们将各个CLB之间或CLB、IOB之间以及IOB之间连接起来，构成特定功能的电路。

FPGA的功能由逻辑结构的配置数据决定。工作时，这些配置数据存放在片内的SRAM或熔丝图上。基于SRAM的FPGA器件，在工作前需要从芯片外部加载配置数据。配置数据可以存储在片外的EPROM、E²PROM或计算机软、硬盘中。人们可以控制加载过程，在现场修改器件的逻辑功能，即所谓现场编程。

可配置逻辑块(CLB)一般有三种结构形式：① 查找表结构；② 多路开关结构；③ 多级与非门结构。不同厂家生产的FPGA，其CLB、IOB等结构都存在较大的差异，这里不再赘述。。

3. 可编程逻辑器件的开发过程

在数字系统的设计中，通常有两种设计思路，一种是自顶向下（Top-Down）的设计，另一种是自底向上（Bottom-Up）的设计。传统的数字系统设计多采用自底向上的方法，基于PLD的数字系统多采用自顶向下的设计，设计者先将一个硬件系统划分成几个大的模块，设计出各大模块的行为功能或结构，并进行仿真以检验设计是否正确，然后将大的模块分给下一级设计者。

一般来说，完整的PLD设计流程包括设计准备、设计输入、设计处理和器件编程四个主要步骤，同时包括相应的功能仿真、时序仿真和器件测试三个设计验证过程。

可编程逻辑器件的基本开发流程如图8-25所示。

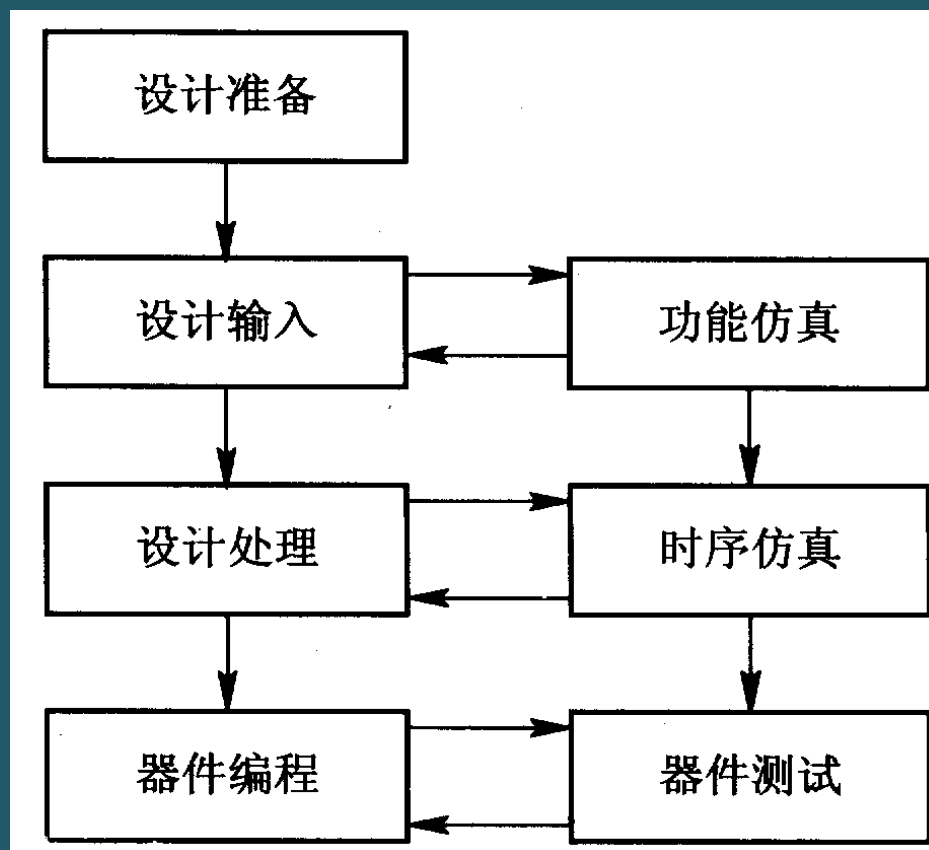


图 8-25 PLD设计流程

1. 设计准备

采用有效的设计方案是PLD设计成功的关键，因此在设计输入之前首先要考虑两个问题：

- ① 选择系统方案，进行抽象的逻辑设计；
- ② 选择合适的器件，满足设计的要求。

2. 设计输入

设计者将所设计的系统或电路以开发软件要求的某种形式表示出来，并送入计算机的过程称为设计输入。它通常有原理图输入、硬件描述语言输入和波形输入等多种方式。



3. 设计处理

从设计输入完成以后到编程文件产生的整个编译，适配过程通常称为设计处理或设计实现。它是器件设计中的核心环节，是由计算机自动完成的，设计者只能通过设置参数来控制其处理过程。在编译过程中，编译软件对设计输入文件进行逻辑化简、综合和优化，并适当地选用一个或多个器件自动进行适配和布局、布线，最后产生编程用的编程文件。

4. 设计校验

设计校验过程包括功能仿真和时序仿真，这两项工作是在设计输入和设计处理过程中同时进行的。

功能仿真是在设计输入完成以后的逻辑功能验证，又称前仿真。它没有延时信息，对于初步功能检测非常方便。时序仿真在选择好器件并完成布局、布线之后进行，又称后仿真或定时仿真。时序仿真可以用来分析系统中各部分的时序关系以及仿真设计性能。

5. 器件编程与调试

设计开发的最后步骤就是在线调试或者将生成的配置文件写入PLD芯片中（称为器件编程）并进行测试。在下载测试以前，应进行器件选择和管脚分配，并重新编译。下载完成后，可以进行静态和动态的电路测试。