

■ 聚焦 EDA

基于 Xilinx ISE 的 FPGA/CPLD 设计与应用

谈世哲 李 健 管殿柱 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

聚焦 EDA

基于 Xilinx ISE 的 FPGA/CPLD 设计与应用

谈世哲 李健 管殿柱 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

FPGA/CPLD 以其强大的功能, 开发过程短, 投资少, 可反复修改, 保密性好, 开发工具智能化等特点, 成为当今硬件设计的首选方式之一。

Xilinx ISE 是由顶级 FPGA 供应商 Xilinx 提供的高级 FPGA/CPLD 设计环境, 为所有 Xilinx 的 FPGA 与 CPLD 提供支持。本书旨在向读者介绍 ISE 强大的开发功能, 通过实例来强化读者的理解和使用。本书立足工程实践, 循序渐进地介绍了 Xilinx ISE 开发的基本过程和方法, 内容翔实、系统、全面, 并通过大量的工程实例说明软件的功能和应用方法。

本书图文并茂, 讲解深入浅出, 通俗易懂。本书适合从事 FPGA/CPLD 设计开发的技术人员阅读, 也可作为高等学校相关专业的教学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

基于 Xilinx ISE 的 FPAG/CPLD 设计与应用 / 谈世哲, 李健, 管殿柱编著. —北京: 电子工业出版社, 2009.8
(聚焦 EDA)

ISBN 978-7-121-09359-3

I. 基… II. ①谈… ②李… ③管… III. 可编程逻辑器件—系统设计 IV. TP332.1

中国版本图书馆 CIP 数据核字 (2009) 第 132199 号

责任编辑: 张 剑 (zhang@phei.com.cn)

特约编辑: 郭 莉

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 15.25 字数: 390 千字

印 次: 2009 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 29.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

Xilinx ISE 9.0 是目前业界最完整的可编程逻辑设计解决方案,用于实现最优性能和功率管理,降低成本,提高生产效率。Xilinx ISE 9.0 可以帮助设计者在更少的时间内实现业内最快速的 FPGA 性能,是 Xilinx 公司最新推出的业内领先的设计工具,其性能比其他解决方案平均快 30%,能够让用户更快、更轻松地完成时序收敛,为 FPGA/CPLD 的开发设计提供了一个完美的平台。

本书以 Xilinx ISE 9.2 为基础,全面介绍了 ISE FPGA/CPLD 的基本功能和应用,包括 VHDL 语言、设计仿真方法及编程下载等。书中运用大量的工程实例对 Xilinx ISE 集成开发环境的基本功能及相关理论知识进行综合运用,以期加强读者对 ISE 的设计应用技能。

本书的写作思想立足于实际问题的应用设计,通过具有针对性和代表性的实例讲解常用工具、命令等,以此开拓读者思路,使其掌握方法,提高综合运用知识的能力。在学习过程中,通过工程实例,循序渐进地练习,使读者真正掌握基于 Xilinx ISE 的 FPGA/CPLD 的设计方法及技巧。

对于已经了解了一些关于 Xilinx ISE 基础知识的读者,本书的相关内容在于澄清概念,提高实践技能,在设计思想和方法方面的探讨对设计人员具有一定的参考意义。

本书由谈世哲、李健、管殿柱编著。参与本书编写的还有田东、李文秋、宋一兵、曹立文、刘平、刘芸、田绪东、付本国、张洪信、宿晓宁、赵水和刘刚。由于知识水平和经验的局限性,书中难免存在缺点和错误,敬请广大读者批评指正。

编 著 者

目 录

第 1 章 聚焦 Xilinx ISE	1
1.1 Xilinx 公司及其产品介绍	1
1.1.1 Xilinx 公司简介	1
1.1.2 几种 CPLD 系列芯片的特点	2
1.1.3 CoolRunner 系列的高级特性	3
1.1.4 主流 FPGA 产品	8
1.2 FPGA/CPLD 基本结构与实现原理	10
1.2.1 FPGA 基本结构与实现原理	10
1.2.2 CPLD 基本结构与实现原理	12
1.2.3 FPGA/CPLD 性能特点差异	14
1.3 系统设计语言——VHDL 基本概念与程序结构	15
1.3.1 概述	15
1.3.2 VHDL 程序基本结构	16
1.4 HDL 编码风格及规则	22
1.4.1 编码风格	22
1.4.2 HDL 编码指导	26
1.5 ISE 基本操作	28
1.5.1 ISE 的获取	28
1.5.2 ISE 的实现功能	29
1.5.3 ISE 软件运行硬件环境及安装	30
1.5.4 ISE 运行及主界面简介	34
1.6 本章小结	37
1.7 思考与练习	37
第 2 章 基于 VHDL 语言的设计输入	38
2.1 新建工程	38
2.2 手动新建源代码	44
2.3 利用语言模板创建源代码	46
2.4 本章小结	49
2.5 思考与练习	49
第 3 章 设计仿真	50
3.1 仿真基本概念	50
3.1.1 仿真类型	50
3.1.2 仿真的步骤	51
3.2 创建测试基准波形文件	56
3.3 使用 Modelsim 进行仿真	62

3.3.1 ModelSim 仿真窗口综述	63
3.3.2 在 ISE 集成环境中进行功能仿真	68
3.3.3 利用 ModelSim 进行时序仿真	71
3.4 本章小结	73
3.5 思考与练习	73
第 4 章 基于原理图与状态机的输入	74
4.1 原理图设计概述	74
4.1.1 顶层原理图设计方法	77
4.1.2 底层原理图设计方式	77
4.2 利用原理图的设计方法	79
4.2.1 自顶向下的原理图设计方法	79
4.2.2 自底向上的原理图设计方法	80
4.3 实例化计数器	80
4.3.1 例化 VHDL 模块	80
4.3.2 进行原理图连线	84
4.3.3 给连线添加网络名	84
4.3.4 给总线添加网络名	86
4.3.5 添加 I/O 引脚标记	87
4.4 状态机输入工具——StateCAD	89
4.4.1 StateCAD 简介	89
4.4.2 StateCAD 用户界面	89
4.4.3 使用 StateCAD 设计状态机	94
4.5 本章小结	106
4.6 思考与练习	106
第 5 章 综合与设计实现	107
5.1 XST 概述	107
5.1.1 XST 属性描述及设置方法	108
5.1.2 XST 操作流程	112
5.2 设计实现	115
5.2.1 CPLD 的设计实现	115
5.2.2 FPGA 的设计实现	118
5.3 约束	120
5.3.1 创建 UCF 文件	120
5.3.2 UCF 文件的语法说明	122
5.3.3 引脚和区域约束语法	123
5.3.4 PACE	125
5.4 IP Core 简介	132
5.4.1 Xilinx IP Core 基本操作	132
5.4.2 DDS 模块 IP Core 的调用实例	136
5.5 本章小结	139
5.6 思考与练习	139

第 6 章 功耗分析与 FPGA/CPLD 配置	140
6.1 功耗评估工具——XPower	140
6.1.1 概述	140
6.1.2 XPower 操作界面	141
6.1.3 功耗分析	144
6.2 基于 ISE 的硬件编程	150
6.2.1 iMPACT 的用户界面	151
6.2.2 利用 iMPACT 进行程序下载	154
6.3 本章小结	157
6.4 思考与练习	158
第 7 章 应用实例	159
7.1 VHDL 数字逻辑电路设计试验	159
7.2 实例一：模可变 16 位加法计数器	159
7.3 实例二：多倍次分频器	170
7.4 实例三：奇偶校验	181
7.5 实例四：数字频率计 VHDL 程序与仿真	192
7.6 实例五：UART VHDL 程序与仿真	199
7.7 实例六：电子时钟 VHDL 程序与仿真	221
7.8 本章小结	234
7.9 思考与练习	235
参考文献	236

第 1 章 聚焦 Xilinx ISE

本章首先介绍了 Xilinx 公司及其产品的基本情况，并在此基础上描述了 CPLD 和 FPGA 的内部结构及基本原理，这将有助于系统设计者深入理解 ISE 软件的设计过程；其次，本章介绍了 VHDL 硬件描述语言基础知识；最后，详细地讲解了设计软件 ISE 的安装过程及其主操作界面。

1.1 Xilinx 公司及其产品介绍

总部设在加利福尼亚圣何塞市（San Jose）的 Xilinx 是全球领先的可编程逻辑解决方案的供应商，图 1-1 为公司标志。Xilinx 公司的业务是研发、制造并销售高级集成电路、软件设计工具以及作为预定义系统级功能的 IP（Intellectual Property）核，其相关产品在全球占有大量的份额，客户通过使用 Xilinx 及其合作伙伴的自动化设计软件和 IP 核，进行器件编程及设计的工作，最终实现特定的逻辑功能。



图1-1 Xilinx公司标志

1.1.1 Xilinx 公司简介

Xilinx 公司成立于 1984 年，它首创了现场可编程逻辑阵列（FPGA）这一创新性的技术，并于 1985 年首次推出商业化产品。目前 Xilinx 满足了全世界对 FPGA 产品一半以上的需求。Xilinx 产品线还包括复杂可编程逻辑器件（CPLD）。Xilinx 可编程逻辑解决方案缩短了电子设备制造商开发产品的时间并加快了产品面市的速度，从而减小了制造商的风险。与采用传统方法，如固定逻辑门阵列相比，利用 Xilinx 可编程器件，客户可以更快地设计和验证他们的电路。而且，由于 Xilinx 器件是只需要进行编程的标准部件，客户不需要像采用固定逻辑芯片时那样等待样品或者付出巨额成本。Xilinx 产品已经被广泛应用于从无线电话基站到 DVD 播放机的数字电子应用技术中。

作为一个可编程逻辑器件的供应商，Xilinx 有自己的开发套件，至今已经形成了一个完整的整体。

Xilinx 公司提供的开发软件主要有以下几种。

- ISE: Xilinx 公司集成开发的工具
- Foundation: Xilinx 公司早期的开发工具，逐步被 ISE 取代
- ISE Webpack: Webpack 是 Xilinx 公司提供的免费开发软件，功能比 ISE 少一些，可以从 Xilinx 网站下载
- 嵌入式开发套件（EDK）：用于开发集成 PowerPC 硬核和 Micro Blaze 软核 CPU 的工具
- System Generator for DSP: 配合 Matlab，在 FPGA 中完成数字信号处理的工具
- Xilinx IP 核: Xilinx 公司拥有非常完整的 IP 库

1.1.2 几种 CPLD 系列芯片的特点

1. XC9500XL 系列 CPLD

该系列整合了先进的架构和有效的制造工艺，为用户提供了巨大的灵活性、可使系统具有较高性能，同时产品具有开发周期较短，成本较低等优点。

图 1-2 所示为 XC9572XL 实物图。该系列的主要设计特性体现在以下几点：



图1-2 XC9572XL实物图

- 具有 5.0V、3.3V 和 2.5V I/O 接口
- 提供 36~288 个宏单元密度，并具有多种封装选项和 I/O 性能，能够很轻松地实现不同密度器件间的移植

- 具有先进的、第二代引脚锁定技术，无需改变板布局即可轻松实现重新设计
- 支持 IEEE 1149.1 JTAG 的边界扫描技术
- 具有快速的在系统编程能力并且擦除时间极短
- 具有可重配置功能，高编程耐久性

2. XC9500XV 系列 CPLD

Xilinx 公司的 XC9500XV 2.5V 系列的 CPLD 采用了先进架构，具有强大的灵活性和低成本特性，可以加快产品上市时间，降低制造与技术支持成本。其主要特点如下：

- 低功耗，具有良好的可靠性
- 能够提供具有良好效益的逻辑解决方案
- 提供了与 3.3V XC9500XL 系列一样的先进架构特性和密度，器件密度有 36、72、144 个和 288 个宏单元
- 支持业界最广泛的 IEEE1149.1 JTAG 和 IEEE 1532 编程接口标准
- 2.5V XC9500XV 器件为在系统编程（ISP）提供了优化的支持

此外，XC9500XV 具有总线保持电路，并提供了良好的 I/O 控制、热插拔功能等，为用户进行系统设计提供较强的灵活性。

3. XC9500 系列 CPLD

XC9500 系列 CPLD 为用户提供了高可靠性，低成本的可编程逻辑解决方案。在保持高性能的同时，XC9500 系列还能为用户提供最大的布线能力和灵活性。XC9500 系列 CPLD 特点包括：

- 具有单个 p-term 输出，其 p-term 输出具有丰富的资源
- 具有 3 个全局时钟
- 具有良好的灵活性，具有较强的适应设计变化的能力
- 具有在系统编程能力，并且操作过程简单
- 每个密度点多提供了 12% 以上的逻辑资源，而且不需要支付额外的成本
- 采用 Flash 工艺

4. CoolRunner-II 系列

CoolRunner-II 系列采用了业界最先进的 CPLD 技术、具有较低的待机功耗和动态功耗、具有良好的系统性能，采用了低成本封装。Xilinx CoolRunner-II 系列特点描述如下：

- 具有 28.8 mW 的超低功耗

- 典型的 16μA 待机电流
- 增加了 DataGate 和先进的 I/O 技术
- 采用了 QF32 和 QF48 小型化低成本的封装解决方案，为业界尺寸最小的封装形式

5. CoolRunner XPLA3 系列 CPLD

该系列为 3.3V CPLD，CoolRunner XPLA3 CPLD 系列具有以下性能：

- 提供了 4.5ns 的性能和 17~18μA 的待机电流，比其他任何 3.3V CPLD 产品都具有更加出色的性能/功率组合
- 主要面向便携式、手持式等低功耗产品的应用
- 采用了 Fast Zero Power (FZP) 设计技术，从而可以在一个可编程逻辑器件中同时实现低功耗和高性能
- 提供了业内首个最有效的 PLD 构架，包括快速输入寄存器；全 PLA 和 VFM 单元
- 具有出色的引脚锁定

1.1.3 CoolRunner 系列的高级特性

本小节将描述 CoolRunner-II 的一些高级特性，包括双边沿触发（DET）寄存器，时钟分频器，CoolCLOCK 技术，DataGATE 功能等，这些特性将有助于用户进行相关的设计，从而提高产品的性能。

1. 双边沿（DualEdge）触发寄存器

CoolRunner-II 双边沿触发寄存器能够大大提高产品的设计性能。双边沿触发（DET）寄存器能够在时钟的上升沿和下降沿完成数据的寄存，从而可以大大地提高系统性能。CoolRunner-II DET 寄存器可以实现的逻辑功能包括移位寄存器、计数器、比较器和状态机。在使用 DET 寄存器之前，用户必须对设计时所需要的 CPLD 逻辑性能进行评估，并可通过采用 ABEL、HDL 或原理图等不同的设计输入方式，来完成 DET 寄存器的设计或例化。表 1-1 列出了可用于创建 CoolRunner-II DET 寄存器的方法。

表 1-1 DET 寄存器的例化方法

设计输入方式	例化实例
ABEL	QOUT: =data; QOUT, DEC=clock
VHDL/Verilog	推断双边沿触发寄存器
原理图设计	例化 FDDn[S][R][E]元件

用户可以在 HDL 设计中完成单边沿触发（SET）寄存器。一个输入时钟上升沿有效单边沿触发寄存器的 VHDL 或 Verilog 语法描述如下：

```
VHDL:if (clock'event) and (clock = '1') then
Verilog:always @ (posedge clock)
```

设计 CoolRunner-II DET 寄存器时，要求寄存器在时钟的上升沿和下降沿上都处于有效状态。

CoolRunner DET 寄存器的 VHDL 范例如下：

```
Process (clock)
begin
  if (clock'event) then
    ...
  end if;
```



```
end process;
```

CoolRunner- II DET 寄存器 Verilog 范例如下:

```
always @(negedge clock or posedge clock)
```

此外，DET 寄存器可供 CoolRunner- II 家族所有器件中的所有宏单元使用。

2. 时钟分频器

1) 时钟分频器的特性 DET 寄存器附加的同步时钟由 CoolRunner- II CPLD 时钟分频器提供，具有良好的灵活性。CoolRunner- II 时钟分频器完成对输入时钟的分频，并将分频时钟全局地分配给所有宏单元。

此外，时钟分频器可以通过减少内部时钟网络的翻转率，来实现额外的省电效果。

CoolRunner- II 时钟分频器可用于全局时钟 GCK2，可对输入时钟进行 2 分频、4 分频、6 分频、8 分频、10 分频、12 分频、14 分频和 16 分频，时钟分频器可以产生 50-50 占空比的分频时钟，并且不影响 TCO。CPLD 在上电时，复位电路将初始化时钟分频器的输出为低电平。

2) 时钟分频器的结构及元件 CoolRunner- II 时钟分频器结构如图 1-3 所示，从图中可以看到，时钟分频器电路包含了一个 CDRST（高电平有效的同步复位）输入信号，当 CDRST 信号输入有效时，时钟分频器在当前周期结束后，停止输出。当 CDRST 信号无效时，时钟分频器将在 GCK2 第一个边沿上开始工作。

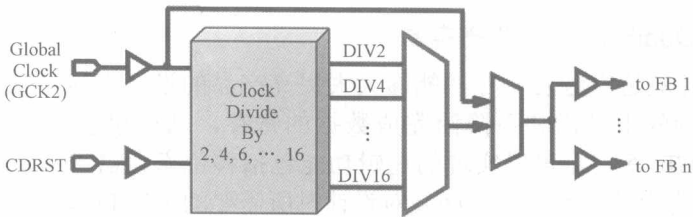


图1-3 CoolRunner- II 时钟分频器结构

CoolRunner- II 时钟分频器还具有内置的延迟电路。当延迟电路的延迟功能启用时，时钟分频器的输出将被延迟一个完整的计数周期。使用延迟功能时，时钟分频器在分频器到达终止计数值之前不会输出上升时钟沿。延迟功能可在配置时启用或禁用。设计时所例化的时钟分频器元件的类型将决定是启用还是禁用延迟功能。图 1-4 描述了启用或禁用延迟功能情况下，CoolRunner- II 时钟分频器的时序波形。

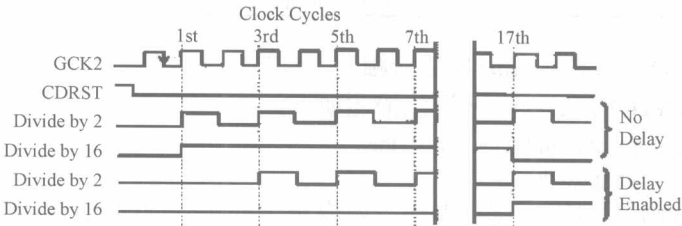


图1-4 CoolRunner- II 时钟分频器的时序波形

Xilinx 的综合技术 (XST) 允许直接在 HDL 源代码中例化时钟分频器元件，也可以采用 ABEL、或原理图设计方法完成时钟分频器元件的例化。表 1-2 列出了可在 ABEL、HDL 或原理图设计中例化的时钟分频器元件。

表 1-2 例化的时钟分频器元件

元 件	描 述
CLK_DIVn	全局时钟分频器元件。不支持同步复位或开始延迟功能。可用: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16
CLK_DIVnR	支持同步复位的全局时钟分频器。不支持开始延迟功能。可用: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16R
CLK_DIVnSD	支持开始延迟的全局时钟分频器。不支持同步复位。可用: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16SD
CLK_DIVnRSD	支持同步复位和开始延迟功能的全局时钟分频器。可用: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16RSD

3) 时钟分频器元件声明和例化 在 VHDL 中, 如果要使用 CoolRunner- II 时钟分频器, 就要进行“元件声明”和“元件例化”的描述。

“元件声明”主要用于定义时钟分频器单元的名称和接口。下面是时钟分频因子为 2 的 CLK_DIV2 元件的 VHDL 元件声明语法。

```
component CLK_DIV2 is
  port ( CLKIN : in STD_LOGIC;
         CLKDV : out STD_LOGIC );
end component;
```

元件例化是将信号与时钟分频器元件的端口进行关联。假设设计需要进行 2 分频, 则必须例化 CLK_DIV2 元件。下面是例化 CLK_DIV2 元件的 VHDL 语法示例, 其中将输入的时钟信号 clk 定义在 CLKIN 端口上, 时钟分频器输出信号 clk_div_by_2 定义在 CLKDV 端口上。

```
U1:CLK_DIV2
  port map(
    CLKIN => clk,
    CLKDV => clk_div_by_2);
```

4) 支持同步复位和延迟功能的时钟分频器元件声明和例化 假设采用分频因子为 16 的时钟分频, 并且支持同步复位和延迟功能, 则必须声明和例化 CLK_DIV16RSD 元件。其元件声明的 VHDL 示例如下:

```
component CLK_DIV16RSD is
  port ( CLKIN : in STD_LOGIC;
         CDRST : in STD_LOGIC;
         CLKDV : out STD_LOGIC );
end component;
```

元件例化功能是分配端口信号。将输入时钟信号 clk 分配在 CLKIN 端口上, 时钟分频器复位信号 clk_div_rst 分配在 CDRST 端口上, 时钟分频器输出 clk_div_by_16 分配在 CLKDV 端口上, 同时根据要求使能 CoolRunner- II 时钟分频器的延迟功能。下面是采用 VHDL 语言例化 CLK_DIV16RSD 元件的示例:

```
U1: CLK_DIV16RSD
  port map (
    CLKIN => clk,
    CDRST => clk_div_rst,
    CLKDV => clk_div_by_16);
```

5) Verilog 设计方法下的时钟分频器元件声明和例化 前面介绍的时钟分频器元件声明和例化方法采用了 VHDL 语言, 下面将介绍采用 Verilog 以及 ABEL 语言的设计范例。

(1) Verilog 设计范例。与 VHDL 设计输入不同, 采用 XST 的 Verilog 设计输入时, 不需要进行元件声明, 只需要进行元件的例化工作。

下面是一个采用 Verilog 语法, 完成 CLK_DIV16RSD 元件例化的示例。在该示例中, 输入时钟信号 clk 分配到 CLKIN 端口, 时钟分频器复位信号 clk_div_rst 分配到 CDRST 端

口, 时钟分频器输出 `clk_div_by_16` 分配到 `CLKDV` 端口, 具体示例描述如下:

```
CLK_DIV16RSD U1 (
    ,CLKIN (clk),
    ,CDRST (clk_div_rst),
    ,CLKDV (clk_div_by_16) );
```

(2) ABEL 设计范例。采用 ABEL 设计输入进行 CoolRunner-II 时钟分频器设计时, 与 VHDL 设计输入一样都需要元件声明和元件例化, 所不同的是, 采用 ABEL 设计输入时, 必须声明时钟分频器为外部元件:

```
CLK_DIV2R external (CLKIN, CDRST -> CLKDV);
```

在元件例化过程中, 首先要为时钟分频器元件定义一个标识符, 这一过程是利用 ABEL 关键字 `functional_block` 来完成的, 下面的示例为时钟分频器元件 `CLK_DIV2R` 定义了一个标识符 `U1`:

```
U1 functional_block CLK_DIV2R;
```

在完成标识符定义后, 就需要实现元件端口的映射。下面的示例将输入时钟 `clk` 映射到 `CLKIN` 端口, 时钟分频器复位信号 `clk_div_rst` 映射到 `CDRST` 端口, 时钟分频器输出信号 `clk_div_by_2` 分配给 `CLKDV` 输出端口, 从而完成端口的映射过程。

```
U1, CLKIN = clk;
U1, CDRST = clk_div_rst;
clk_div_by_2 = U1, CLKDV;
```

3. CoolCLOCK 技术

CoolRunner-II 的 CoolCLOCK 技术是一种组合全局时钟分频器和 DET 寄存器功能的技术。该技术通过将全局时钟进行 2 分频, 使内部时钟网络上分配更低频率的时钟, 然后在各个宏单元上, 再将时钟加倍, 从而达到降低功耗的目的。由于时钟分频器和 DET 寄存器的零插入延迟特性, 可获得零时钟偏斜。图 1-5 为 CoolRunner-II CoolCLOCK 特性。

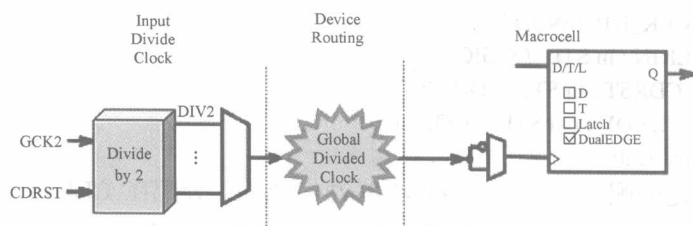


图1-5 CoolRunner-II CoolCLOCK特性

由于 GCK2 是唯一的可分频时钟网络, 因此 CoolCLOCK 功能只能在 GCK2 网络上使用。通过定义输入时钟的属性, 来设置 CoolCLOCK 特性, 换句话说, CoolCLOCK 功能可通过设置输入时钟的属性来实现。通过设置 CoolCLOCK 属性, 可以免除实例化时钟分频器和 DET 寄存器的需要。表 1-3 列出了 CoolCLOCK 属性, 设置语法和设置范例。

表 1-3 CoolCLOCK 属性

设计输入方式	语 法	范 例
UCF	NET <时钟名称> COOL_CLK	NET clk COOL_CLK
ABEL	XILINX PROPERTY 'COOL_CLK <时钟名称>;	XILINX PROPERTY 'COOL_CLK clk';
VHDL	attribute COOL_CLK : string; attribute COOL_CLK of <时钟名称>: signal is "TRUE";	attribute COOL_CLK : string; attribute COOL_CLK of clk : signal is "TRUE";

续表

设计输入方式	语 法	范 例
Verilog	//SYNTHESIS attribute COOL_CLK of <时钟名称>: signal is "TRUE"; 注意: 注释定界符是有意义的, 对于 XST 必不可少	//SYNTHESIS attribute COOL_CLK of clk : signal is "TRUE";

注意: CoolCLOCK 可在 CoolRunner-II 128 宏单元器件和更大的器件上使用。

4. DataGATE 特性

在实际设计过程中, 往往需要阻塞特定的输入, 通过阻塞输入, 转换信号不会驱动内部电容, 从而减少总功耗。CoolRunner-II 提供了 DataGATE 特性来实现这一功能。在 DataGATE 有效之前, 被阻塞的输入引脚上的最后一个值会被锁存, 以供 CPLD 内部使用。图 1-6 描述了 DataGATE 功能的基本原理。

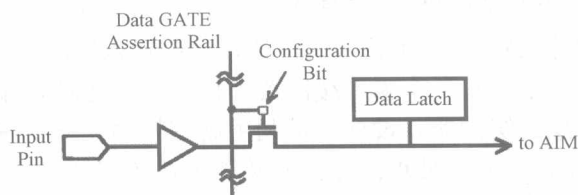


图1-6 DataGATE功能的基本原理

在 CoolRunner-II 中, 有两种属性与 DataGATE 特性相关联。第一种属性定义了输入是否受到 DataGATE 的影响, 第二种属性定义用于指定 DataGATE 控制信号。

设计人员可以逐个引脚选择 DataGATE 特性。当输入引脚需要设置具有 DataGATE 功能时, 可以通过定义该输入引脚为 DATA_GATE 属性来实现。表 1-4 描述了不同设计输入方法下, 对输入信号设置 DataGATE 功能的语法和范例

表 1-4 设置 DataGATE 功能的语法和范例

设计输入方式	语 法	范 例
UCF	NET <信号名称> DATA_GATE;	NET data_in DATA_GATE;
ABEL	XILINX PROPERTY'DATA_GATE <信号名称>;	XILINX PROPERTY'DATA_GATE data_in';
VHDL	attribute DATA_GATE : STRING; attribute DATA_GATE of <信号名称>: signal is "TRUE"; 注意: 只需要对所有 DATA_GATE 属性声明一次字符串属性	attribute DATA_GATE : STRING; attribute DATA_GATE of data_in : signal is "TRUE";
Verilog	//SYNTHESIS attribute DATA_GATE of <信号名称> : signal is "TRUE"; 注意: 注释定界符是有意义的, 对于 XST 必不可少	//SYNTHESIS attribute DATA_GATE of data_in : signal is "TRUE";

通过使用 I/O 引脚或内部逻辑可以完成对 DataGATE 的驱动控制。对于每一个 CoolRunner-II 封装, DataGATE 的使能信号来自专用的 DGE/I/O 引脚。

软件识别 DataGATE 的设计并自动分配 I/O 引脚给 DataGATE 以用做使能控制功能信号——DGE。通过 BUFG DATA_GATE 属性定义和分配 I/O 引脚, 以生成内部 DataGATE 控制逻辑。表 1-5 描述了定义和分配 DataGATE 使能信号的语法和方法。



表 1-5 定义和分配 DataGATE 使能信号的语法和方法

设计输入方式	语 法	范 例
UCF	NET <信号名称> BUFG=DATA_GATE;	NET dg_en BUFG=DATA_GATE;
ABEL	XILINX PROPERTY 'BUFG=DATA_GATE <信号名称>;	XILINX PROPERTY 'BUFG=DATA_GATE dg_en';
VHDL	attribute BUFG : STRING; attribute BUFG of <信号名称>: signal is "DATA_GATE"; 注意：只需要对所有 BUFG 属性声明一次串属性	attribute BUFG : STRING; attribute BUFG of dg_en : signal is "DATA_GATE";
Verilog	//SYNTHESIS attribute BUFG of <信号名> : signal is "DATA_GATE"; 注意：注释定界符是有意义的，对于 XST 必不可少	//SYNTHESIS attribute BUFG of dg_en : signal is "DATA_GATE";

5. 施密特触发输入缓冲器和 I/O 端接模式

CoolRunner- II 的每个 I/O 有多个输入缓冲器，这些输入缓冲器可以被用于不同的 I/O 配置。在这些输入缓冲器中，有一个输入缓冲器可作为施密特触发输入缓冲器，在 CPLD 配置时，可以启用该类型的缓冲器。设计人员利用施密特触发输入缓冲器，能够在同一器件上灵活地使用 CoolRunner- II，处理高速信号及缓慢转换信号。传统上，当为 CMOS 输入时，缓慢转换信号会造成双时钟或低频干扰，从而导致数字系统大为混乱。这一问题可以通过利用施密特触发器输入缓冲器来有效消除，而使用施密特触发器输入缓冲器的代价只是几纳秒延迟。表 1-6 描述了用于给特定信号进行施密特触发器输入缓冲器分配或定义的语法和范例。

表 1-6 分配施密特触发器输入缓冲器

设计输入方式	语 法	范 例
UCF	NET <信号名称> SCHMITT_TRIGGER;	NET data_in SCHMITT_TRIGGER; NET clock SCHMITT_TRIGGER;
ABEL	XILINX PROPERTY 'SCHMITT_TRIGGER <信号名称>;	XILINX PROPERTY 'SCHMITT_TRIGGER data_in'; XILINX PROPERTY 'SCHMITT_TRIGGER clock';
VHDL	attribute SCHMITT_TRIGGER : STRING; attribute SCHMITT_TRIGGER of <信号名称>: signal is "TRUE"; 注意：只需要对所有 SCHMITT_TRIGGER 属性声明一次串属性	attribute SCHMITT_TRIGGER : STRING; attribute SCHMITT_TRIGGER of data_in : signal is "TRUE"; attribute SCHMITT_TRIGGER of clock : signal is "TRUE";
Verilog	//SYNTHESIS attribute SCHMITT_TRIGGER of <信号名称>; 注意：注释定界符是有意义的，对于 XST 必不可少	//SYNTHESIS attribute SCHMITT_TRIGGER of data_in; //SYNTHESIS attribute SCHMITT_TRIGGER of clock;

1.1.4 主流 FPGA 产品

Xilinx 作为全球领先的可编程逻辑解决方案的供应商，FPGA 产品占据市场一定数量的份额，其主流 FPGA 产品包括 Spartan-3 系列，Virtex-II 系列等，下面进行简单的描述。

1. Spartan-3 系列

Spartan-3 是新一代 FPGA 产品，结构与 Virtex-II 类似，主要型号包括 Spartan-3/3L，

Spartan-3A, Spartan-3E 等, 其主要特点描述如下:

- Spartan-3/3L: 全球第一款 90nm 工艺 FPGA, 1.2V 内核, 于 2003 年开始陆续推出。成本低廉, 总体性能指标不是很优秀, 适合低成本应用场合, 是 Xilinx 未来几年在低端 FPGA 市场上的主要产品, 目前市场上中低容量型号很容易购买到, 容量相对少一些
- Spartan-3A: 适合于对逻辑密度要求比较高的应用场合, 当应用需要有桥接、差分信号和存储器接口等时, 该型号是一种理想的选择
- Spartan-3E: 为 Xilinx 最新推出的低成本 FPGA, 其基本框架是基于 Spartan-3/3L 的, 并在此基础上对性能和成本做了进一步优化, 适合于逻辑密度要求比较高的应用场合, 如应用于逻辑集成、DSP 协处理和嵌入式控制等应用; 该款产品成本低廉, 适合于低成本应用场合, 是 Xilinx 未来几年在低端 FPGA 市场上的主要产品

2. Virtex-II 系列 FPGA

Virtex-II 系列是在 2002 年推出, 采用 0.15 μ m 工艺, 1.5V 内核, 适用于大规模高端 FPGA 产品。Xilinx 为利用 Virtex-II FPGA 进行高性能系统设计提供了所需的全部工具, 提供了世界级的 IP 核解决方案、验证工具以及开发所需要的信息资源, 以确保整个设计开发过程。目前, 该系列的 FPGA 主要包括 Virtex-II、pro Virtex-4、Virtex-5 等型号, 下面做一个简单描述:

- Virtex-II pro: 基于 Virtex-II 的结构, 采用 0.13 μ m、1.5V 工艺技术制造, 整合了嵌入式 PowerPC 处理器和 3.125 Gbps RocketIO 串行收发器的 FPGA 产品, 基于系统内可配置的 SRAM, 具有灵活的逻辑资源; 高达 88192 个内部时钟可控的寄存器、锁存器; 支持多路选择和输入功能; 高性能的时钟管理电路, 高达 12 个数字时钟管理模块; 灵活的频率合成技术; 具有 XCITE 数控阻抗 I/O 端口等
- Virtex-4: 是 Xilinx 最新一代高端 FPGA 产品, 包含三个子系列: LX、SX、FX, Virtex-4 系列于 2004 年推出, 采用 1.2V、90nm、三栅极氧化层技术制造而成, 提供了业界最高的性能, 并极大地降低了功耗。各项指标比上一代 Virtex-II 均有很大提高, 获得 2005 年 EDN 杂志最佳产品称号, 从 2005 年年底开始批量生产, 是未来几年 Xilinx 在高端 FPGA 市场中的最重要的产品
- Virtex-5: 世界上首款采用 1.0V、三栅极氧化层工艺技术制造而成的 65nm 的最新 FPGA 系列产品。Virtex-5 FPGA 的逻辑单元高达 330000 个; I/O 引脚数量高达 1200 个; 此外还带有低功耗 Rocket IO 串行收发器、内置式 PCI Express 端点、以太网 MAC 模块以及其他增强型 IP; 提供了丰富的集成系统性能, 从而缩短了设计周期, 并削减了系统成本。在嵌入式处理方面, 该系列的产品利用丰富和完备的开发工具, 可支持的软处理器核, 从而可以完成构建嵌入式系统或实现复杂的控制功能; Virtex-5 提供了众多的软 IP 核选择, 包括 32bit MicroBlaze 处理器、8bit PicoBlaze 控制器、IBM CoreConnect 总线以及由 Xilinx 及其合作伙伴开发的外围部件。在数字信号处理 (DSP) 方面, Virtex-5 使用增强型 DSP48E slice 可以解决多通道、高性能 DSP 难题, 可用于组建实时视频、成像、无线和加密系统。除此以外, Virtex-5 FPGA 还可以广泛地应用在网络、电信、存储、服务器、计算、无线、广播、视频、成像、医疗、工业和军事应用领域, 以代替 ASIC 和 ASSP

1.2 FPGA/CPLD 基本结构与实现原理

FPGA 和 CPLD 都是可编程 ASIC 器件, 有很多共同特点, 但由于 FPGA 和 CPLD 结构上的差异, 具有各自的特点, 为了更好地理解 FPGA 和 CPLD 的特点和应用场合, 下面将对 FPGA 和 CPLD 基本结构与实现原理进行简单的介绍, 这部分内容将同样有助于用户进行相关的产品设计。

1.2.1 FPGA 基本结构与实现原理

如图 1-7 所示为 FPGA 的内部结构图。

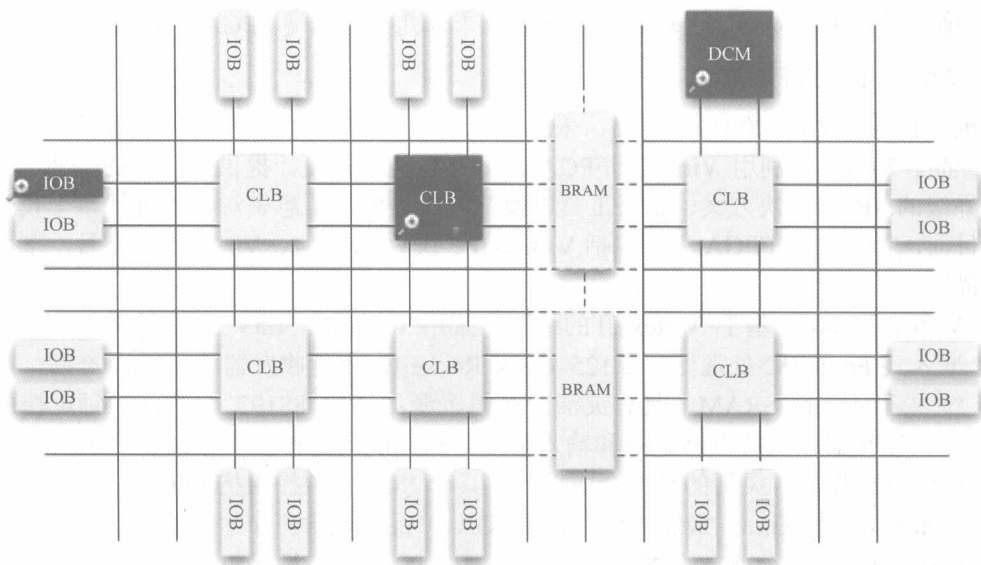


图1-7 FPGA的内部结构图

1. FPGA 基本结构

从图 1-7 中可以看出, FPGA 内部包括 5 个主要组成部分。

1) 可配置逻辑块 (CLB) CLB 是 FPGA 内的基本逻辑单元。实际数量和特性会依器件的不同而不同, 但是每个 CLB 都包含一个由 4 或 6 个输入、一些选型电路 (多路复用器等) 和触发器组成的可配置开关矩阵。开关矩阵是高度灵活的, 可以进行配置以便处理组合逻辑、移位寄存器或 RAM。相应器件的数据手册中提供了更系统的详情。

2) 互连 CLB 提供了逻辑性能, 灵活的互连布线在 CLB 和 I/O 之间发送信号。有几种布线方法, 从专门实现 CLB 互连的到快速水平和垂直长线, 再到实现时钟与其他全局信号的低歪斜发送的器件。除非特别规定, 设计软件使得互连布线任务从用户眼前消失, 这样就极大地降低了设计复杂度。

3) I/O 块 (IOB) 当今的 FPGA 支持很多 I/O 标准, 这样就为您的系统提供了理想的接口连接。FPGA 内的 I/O 按组分类, 每组都能够独立的支持不同的 I/O 标准。当今领先的 FPGA 提供了很多 I/O 组, 这样就实现了 I/O 支持的灵活性。

排成阵列的逻辑单元由布线通道中的可编程内连线连接起来实现一定的逻辑功能, 即分段的金属互连线可以由编程开关以任意方式连接形成逻辑单元之间要求的信号线。