



XILINX大学合作计划指定教材

# FPGA数字信号处理设计教程

## —System Generator入门与提高

■ 纪志成 高春能 编著

# XILINX



西安电子科技大学出版社  
<http://www.xduph.com>



## 内 容 简 介

本书主要介绍基于 FPGA 数字信号处理的设计流程, 探讨数字信号处理算法在 FPGA 中的硬件设计与实现, 重点讲述基于 System Generator 的 FPGA 开发及其构成模块、图形化工程设计流程及实现。本书立足于实践, 结合作者多年从事 FPGA 数字信号处理的设计和教学经验, 通过大量设计实例详细探讨了数字信号处理算法在 FPGA 硬件开发中的详细设计流程(光盘内附详细实例)。所有实例均在 XILINX 公司大学计划 Spartan-3E 开发板上验证通过。

本书可作为使用 XILINX 产品开发数字信号处理系统的工程技术人员及项目管理人员等的参考书, 也可作为有志于数字电路设计的高等院校高年级本科生和研究生的专业教材。

### 图书在版编目 (CIP) 数据

FPGA 数字信号处理设计教程: System Generator 入门与提高 / 纪志成等编著.

—西安: 西安电子科技大学出版社, 2008.2

XILINX 大学合作计划指定教材

ISBN 978-7-5606-1968-2

I. F… II. 纪… III. 数字信号—信号处理—应用软件, System Generator—教材

IV. TN911.72

中国版本图书馆 CIP 数据核字 (2008) 第 000056 号

策 划 戚文艳

责任编辑 戚文艳

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http: //www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 12.75

字 数 295 千字

印 数 1~4000 册

定 价 26.00 元(含光盘)

ISBN 978-7-5606-1968-2/TN · 0403

**XDUP 2260001-1**

\*\*\* 如有印装问题可调换 \*\*\*

本社图书封面为激光防伪覆膜, 谨防盗版。

# 前 言

数字信号处理是利用计算机或专用处理器设备，以数值计算的方法对信号进行采集、变换、综合、估值与识别等加工处理，借以达到提取信息和便于应用的目的。随着半导体集成电路和计算机技术的迅速发展，数字信号处理的理论和技术已经应用到社会的各个方面，成为整个数字化技术的基础。随着系统向高速度、低功耗、低电压和多媒体、网络化、移动化方向的发展，数字信号处理对系统的要求越来越高，传统的基于 DSP 处理器芯片的数字信号处理系统技术难以满足日益提高的整机系统的要求。而现场可编程门阵列(FPGA)由于其固有的特征，正好填补了这方面的要求。FPGA 可以方便地通过对逻辑结构的修改和配置，完成对系统和设备的升级，以便迅速适应市场的变化。

在以 FPGA 为核心器件设计信号处理系统时，使用 XILINX System Generator for DSP 可在 MATLAB/Simulink 的环境下建立数字信号处理系统的抽象算法，并将抽象算法转化成可靠的硬件实现，它是数字信号处理系统设计与 XILINX FPGA 实现之间的“桥梁”。

XILINX System Generator for DSP 软件是由 XILINX 公司开发的 MATLAB/Simulink 环境下的一个工具箱，在使用 FPGA 作为原型平台运行算法时，这一新软件不仅能够对硬件的真实情况进行仿真，还能够自动生成硬件实现时所需的硬件描述语言代码。与语言设计方法相比，使用 XILINX System Generator for DSP 有三个主要的优势：第一，图形化操作，简单易用；第二，实现的算法能确保与仿真结果相符；第三，无需为仿真和实现建立不同的模型。

本书的目的之一是告诉读者基于 FPGA 的数字信号处理的设计流程；之二是以实现的形式帮助读者快速上手使用 XILINX System Generator for DSP 进行 XILINX FPGA 的开发，本书中所有实验都是针对 XILINX Spartan-3E 开发板设计的。

本书假定读者已经熟悉 MATLAB 的基础知识，由于有许多有关 MATLAB 方面的辅导性教材或参考指南都可利用，因而这些内容本书未予包括。

## 内容安排

本书共分 6 章内容，如果读者是一位彻底的求知者，可以全面研读所有 6 章内容。如果读者时间紧张，可以根据个人不同的知识结构阅读其中的部分章节。在下面的“适合读者”的标签下给出了一些建议，针对不同的读者，推荐阅读本书的部分章节。

### 第 1 章 FPGA 硬件结构

本章回顾了 FPGA 的发展史，重点介绍了 XILINX FPGA 的内部结构及几个重要硬件功能模块，包含器件参数列表和选型指南。

## 第2章 数字信号处理的基本知识

本章回顾了数字信号处理的几个基本知识点，作为本书所用知识点的补充，包括 A/D 和 D/A 转换、傅立叶变换、FIR 滤波器等。对此部分内容比较熟悉的读者可以略过本章。

## 第3章 System Generator 概述

本章介绍了 XILINX System Generator for DSP，包括软件的运行环境和重要特征。

## 第4章 System Generator 库的构成

本章介绍了 System Generator 工具箱中的各个模块，读者可以将其视为技术手册使用，当需要用到某个模块时具体查阅。

## 第5章 图形化工程设计流程及实现

本章对比常规 FPGA 开发流程和基于 System Generator 进行 FPGA 开发的流程，提出了基于 System Generator 的图形化工程设计流程。

## 第6章 应用实例

本章给出了几个使用 System Generator 开发的应用实例，包括数字振荡器、FIR 滤波器、CORDIC 算法和时延数字正切锁相环。

本书中给出的大量实例均由江南大学电气自动化研究所研究生在 MATLAB v7.1.0/Simulink v6.3 (R14.3) Service Pack 3、System Generator 8.1 SP3 和 ISE 8.2 SP3 开发环境下验证通过，并以 XILINX 大学计划 Spartan-3E 开发板作为实验平台。读者在完成本书中的实验时，请先将本书附赠的光盘中的 SG 文件夹拷贝到 PC 机的 C 盘根目录下，SG 文件夹下 solutions 文件夹中的内容为在正确完成实验后应当得到的工程文件。在阅读本书和使用 System Generator 的过程中需要注意的地方和一些小技巧在书中都用“◇”符号标记出。

参与本书编写工作的主要人员有高春能、吴定会、何炯、陈星、梁薇等。全书由纪志成教授负责规划、内容安排、定稿与修改。

## 适合读者

尽管希望读者能完整阅读本书，以获得本书中提供的大量信息，但可能读者没有那么多时间，如果读者希望只阅读对自身有用的章节，这里有一些建议。

### 未接触过 FPGA 的信号处理系统开发人员

建议详细阅读本书除第 2 章外的所有章节，以便能够全面了解 XILINX FPGA 器件并使用 System Generator 进行数字信号处理系统的开发。

### 希望迅速上手 System Generator 的 FPGA 工程开发人员

建议详细阅读本书第 3、5、6 章内容，尤其是其中的实验部分，以便对使用 System Generator 开发 FPGA 的流程有相当的了解。

### 项目管理人员

建议详细阅读本书第 5 章内容，熟悉 FPGA 数字信号处理的设计流程，从而更好地管

理工程项目，保证项目开发的收敛性。

### 希望验证算法的科研人员

建议详细阅读本书第 1、3、5、6 章内容，在 System Generator 中构建数学仿真模型后，利用 System Generator 的强大功能迅速转换成 FPGA 硬件实现，从而验证数学算法的可行性。

## 致谢

---

首先感谢江南大学电气自动化研究所的广大教员和研究生对本书的大力支持，每一章节和实例教程都渗透了他们辛勤的汗水和不倦的工作，同时也感谢所有参与原始资料翻译工作的学生们。

感谢 XILINX 公司中国区大学计划经理谢凯年博士对本书的关心和支持。

本书在写作过程中参考了大量相关文献，在此对这些文献的作者表示深深的感谢！

由于作者水平有限，书中难免有疏忽、不恰当，甚至错误之处，恳请各位读者批评指正。

编 者

2007 年 11 月

# 目 录

|   |    |
|---|----|
| <b>第 1 章 FPGA 硬件结构</b> .....                        | 1  |
| 1.1 FPGA 的可编程技术.....                                | 1  |
| 1.1.1 基于 SRAM 的 FPGA 器件.....                        | 1  |
| 1.1.2 反熔丝 FPGA.....                                 | 2  |
| 1.1.3 基于 Flash 的 FPGA.....                          | 2  |
| 1.2 FPGA 的内部结构.....                                 | 2  |
| 1.2.1 可配置逻辑模块(CLB).....                             | 2  |
| 1.2.2 可配置 I/O 模块.....                               | 5  |
| 1.2.3 块存储器(Block RAM).....                          | 6  |
| 1.2.4 数字时钟管理器(DCM).....                             | 10 |
| 1.2.5 硬件乘法器模块(Multiplier).....                      | 12 |
| 1.3 XILINX 公司主流产品介绍与器件选择.....                       | 12 |
| 1.3.1 主流 FPGA 产品.....                               | 12 |
| 1.3.2 器件的选择.....                                    | 15 |
| 1.4 本章小结.....                                       | 16 |
| <b>第 2 章 数字信号处理的基本知识</b> .....                      | 17 |
| 2.1 模拟/数字转换和数字/模拟转换.....                            | 17 |
| 2.1.1 模拟/数字转换.....                                  | 18 |
| 2.1.2 数字/模拟转换.....                                  | 20 |
| 2.2 离散傅立叶变换(DFT)与快速傅立叶变换(FFT).....                  | 21 |
| 2.2.1 离散傅立叶变换(Discrete Fourier Transform, DFT)..... | 21 |
| 2.2.2 快速傅立叶变换(Fast Fourier Transform, FFT).....     | 22 |
| 2.3 滤波器.....  | 27 |
| 2.3.1 无限脉冲响应数字滤波器(IIR).....                         | 27 |
| 2.3.2 有限脉冲响应数字滤波器(FIR).....                         | 31 |
| 2.3.3 IIR 滤波器与 FIR 滤波器的比较.....                      | 32 |
| 2.4 本章小结.....                                       | 32 |
| <b>第 3 章 System Generator 概述</b> .....              | 33 |
| 3.1 软件需求.....                                       | 33 |
| 3.2 软件安装.....                                       | 34 |
| 3.3 编译 XILINX 硬件描述语言库.....                          | 34 |
| 3.4 FPGA 器件需求.....                                  | 35 |

|  |            |
|--|------------|
| 3.5 使用 FPGA 进行数字信号处理的优势 .....                        | 36         |
| 3.6 用 System Generator 进行系统级建模 .....                 | 39         |
| 3.6.1 MATLAB、Simulink 和 System Generator 的运行环境 ..... | 40         |
| 3.6.2 一般流程 .....                                     | 41         |
| 3.6.3 流程范例 .....                                     | 42         |
| 3.6.4 几个重要特点 .....                                   | 48         |
| 3.7 本章小节 .....                                       | 51         |
| <b>第 4 章 System Generator 库的构成</b> .....             | <b>52</b>  |
| 4.1 System Generator 模块定义 .....                      | 52         |
| 4.2 在 Simulink 模型中引用 XILINX 模块 .....                 | 52         |
| 4.3 XILINX Blockset 库 .....                          | 53         |
| 4.4 XILINX Blockset 库 .....                          | 61         |
| 4.5 XILINX Reference Blockset 库 .....                | 133        |
| 4.6 XILINX XtremeDSP Kit 库 .....                     | 135        |
| 4.7 本章小结 .....                                       | 137        |
| <b>第 5 章 图形化工程设计流程及实现</b> .....                      | <b>138</b> |
| 5.1 常规设计流程 .....                                     | 138        |
| 5.1.1 使用硬件描述语言的 FPGA 开发流程 .....                      | 138        |
| 5.1.2 使用 XILINX CORE Generator 的 FPGA 开发流程 .....     | 139        |
| 5.2 使用 XILINX System Generator 的 FPGA 开发流程 .....     | 141        |
| 5.3 System Generator 的重要功能 .....                     | 144        |
| 5.3.1 硬件描述语言协同仿真 .....                               | 144        |
| 5.3.2 硬件验证 .....                                     | 148        |
| 5.3.3 系统在线调试 .....                                   | 156        |
| 5.3.4 资源估计 .....                                     | 163        |
| 5.4 本章小结 .....                                       | 164        |
| <b>第 6 章 应用实例</b> .....                              | <b>166</b> |
| 6.1 数字振荡器 .....                                      | 166        |
| 6.1.1 用 IIR 滤波器实现振荡器 .....                           | 166        |
| 6.1.2 用查表法实现数控振荡器 .....                              | 171        |
| 6.2 有限脉冲响应(FIR)滤波器 .....                             | 176        |
| 6.2.1 标准 FIR 滤波器 .....                               | 177        |
| 6.2.2 标准 FIR 滤波器的改进结构 .....                          | 178        |
| 6.2.3 转置 4 抽头 FIR 滤波器 .....                          | 179        |
| 6.2.4 转置 4 抽头的 FIR 滤波器的变换结构 .....                    | 179        |
| 6.2.5 使用 System Generator 现有乘加 FIR 模块 .....          | 180        |
| 6.3 CORDIC 算法的原理与运用 .....                            | 182        |
| 6.3.1 CORDIC 算法介绍 .....                              | 182        |
| 6.3.2 CORDIC 算法的运用 .....                             | 184        |



|                        |     |
|------------------------|-----|
| 6.4 时延数字正切锁相环.....     | 188 |
| 6.4.1 零阶时延数字正切锁相环..... | 189 |
| 6.4.2 一阶时延数字正切锁相环..... | 191 |
| 6.5 本章小结.....          | 192 |
| 附录 光盘内容树状图.....        | 193 |
| 参考文献.....              | 194 |

# 第1章 FPGA 硬件结构

FPGA(Field Programmable Gate Array)是一类高集成度的可编程逻辑器件,起源于美国的 XILINX 公司,该公司于 1985 年推出了世界上第一块 FPGA 芯片。在这二十多年的发展过程中,FPGA 的硬件体系结构和软件开发工具都在不断地完善且日趋成熟。从最初的 1200 个可用门,到 20 世纪 90 年代时几十万个可用门,到目前数百万门至上千万门的单片 FPGA 芯片,XILINX、Altera 等世界 FPGA 顶级厂商已经将 FPGA 器件的集成度提高到一个新的水平。FPGA 技术结合了微电子技术、电路技术、EDA 技术,使设计者可以集中精力进行所需逻辑功能的设计。与专用集成电路 ASIC(Application Specific Integrated Circuit)相比,FPGA 具有灵活性高、设计周期短、成本低、风险小等优势,因而得到了广泛应用,同时和 FPGA 相关的各项技术也迅速发展起来。

FPGA 是现场可编程门阵列的简称。之所以赋予这个名字,一方面是因为 FPGA 可以用编写代码的方法实现复杂逻辑电路,另一方面是因为其内部是由规则的逻辑阵列所组成的,非常类似一个由逻辑门构成的阵列,而且这种结构很适合实现复杂设计并可重复编程。

用软件编程的思想实现复杂硬件逻辑设计是 FPGA 的一大特点。FPGA 如何能做到通过软件编程来实现硬件逻辑电路的呢?本章将通过简单介绍 FPGA 内部结构使读者能够初步理解 FPGA 可编程的基本原理。

## 1.1 FPGA 的可编程技术

目前,市场上有三种基本的 FPGA 编程技术:SRAM、反熔丝和 Flash。其中,基于 SRAM 的 FPGA 是迄今为止应用范围最广的架构,主要因为它速度快且具有可重编程能力;反熔丝 FPGA 只具有一次可编程(One Time Programmable, OTP)能力;基于 Flash 的 FPGA 是 FPGA 领域中比较新的技术,也具有重编程的能力。另外,基于 SRAM 的 FPGA 器件经常带来一些额外的成本,包括启动 PROM 支持安全和保密应用的备用电池等。基于反熔丝和 Flash 的 FPGA 没有这些隐含成本,因此可保证有较低的总系统成本。

### 1.1.1 基于 SRAM 的 FPGA 器件

这类产品是基于 SRAM 结构的可再配置型器件,上电时要将配置数据读入片内 SRAM 中,配置完成就可进入工作状态。掉电后 SRAM 中的配置数据丢失,FPGA 内部逻辑关系随之消失。这种基于 SRAM 的 FPGA 可以反复重新编程,这使得系统的“在线升级”变得非常容易。

## 1.1.2 反熔丝 FPGA

反熔丝 FPGA 内部具有反熔丝阵列开关结构，其逻辑功能的定义由专用编程器根据设计实现所给出的数据文件，对其内部的反熔丝阵列开关进行烧录，烧录后形成实际的逻辑电路。这种器件的缺点是只能一次可编程；优点是具有高抗干扰性和低功耗，适合于要求高可靠性、高保密性的定型产品。

## 1.1.3 基于 Flash 的 FPGA

在这类 FPGA 器件中集成了 SRAM 和非易失性 EEPROM 两类存储结构。其中 SRAM 用于在器件正常工作时对系统进行控制，而 EEPROM 则用来配置 SRAM。由于这类 FPGA 将 EEPROM 集成在基于 SRAM 工艺的现场可编程器件中，因而可以充分发挥 EEPROM 的非易失特性和 SRAM 的重配置性。掉电后，配置信息保存在片内的 EEPROM 中，因此不需要片外的配置芯片，有助于降低系统成本、提高设计的安全性。

# 1.2 FPGA 的内部结构

每一个 FPGA 的生产厂商都有自己的 FPGA 内部结构体系，但各个厂商设计的基本原理都大同小异。下面以 XILINX 公司的产品为例介绍 FPGA 的内部结构。如图 1.2.1 所示为 XILINX 公司一典型 FPGA 的内部基本结构，这一结构由可配置逻辑模块(CLB)、可配置 I/O 模块、块存储器(Block RAM)以及数字时钟管理器(DCM)和乘法器模块(Multiplier)等基本模块构成。

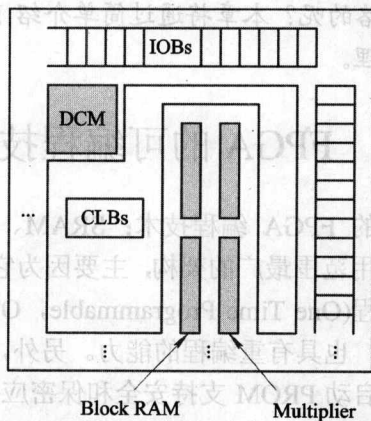


图 1.2.1 XILINX 公司 FPGA 基本结构

### 1.2.1 可配置逻辑模块(CLB)

可配置逻辑模块(CLB)包含了 FPGA 的可编程逻辑。如图 1.2.2 所示为 XILINX 公司 Spartan-3 系列一个典型的 CLB 结构，它由 4 个 Slice 和附加逻辑构成，用于实现组合逻辑和时序逻辑。4 个 Slice 通过一个内部互联线实现内部互联并与相邻的 CLB 连接。开关矩阵用来传递 CLB 中 Slice 的使能信号。其中右边两个 Slice 为逻辑型，简称 Slicel。其内部包

含了寄存器、进位逻辑、查找表和算术逻辑。左边两个 Slice 则为存储型，简称 Slicem。其内部除了具有逻辑型 Slice 所有结构外，还增加了基于查找表的 16×1 位分布式存储器 RAM 和 16 位的移位寄存器。

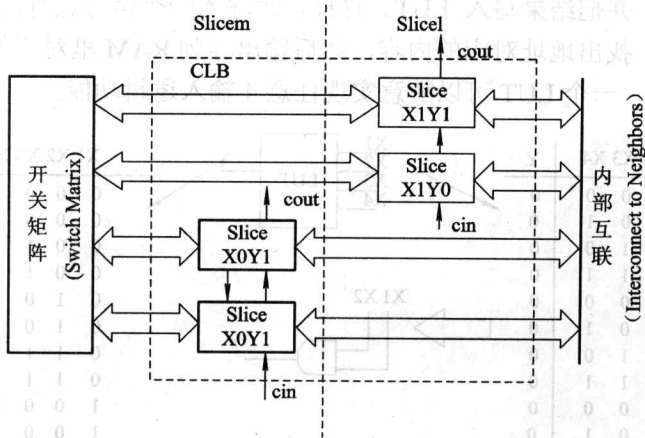


图 1.2.2 CLB 结构图

CLB 细分后的 Slice 模块中包含有 4 输入函数发生器，可以用于实现 4 输入查找表 (LUT)、分布式存储器 RAM 和 16 位基于查找表的移位寄存器(SRL16)，Slice 内部结构如图 1.2.3 所示。时序逻辑可配置为 D 触发器或锁存器。进位逻辑包括两条快速进位链，用于提高 CLB 模块的算术处理速度。算术逻辑包括一个异或门和一个加速乘法运算的“乘累加”逻辑门。每个 CLB 模块既可以配置成分布式 ROM，也可以配置成分布式 RAM。

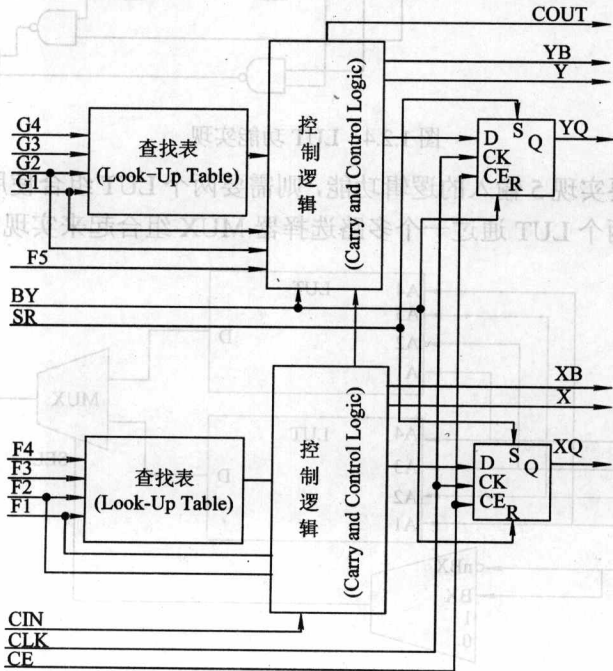


图 1.2.3 Slice 结构图

查找表(Look-Up Table)简称为 LUT，LUT 本质上就是一个可配置的 RAM。目前 FPGA 中大多使用 4 输入的 LUT，所以每一个 LUT 可以看成是一个有 4 位地址线的  $16 \times 1$  的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后，EDA 软件会自动计算逻辑电路的所有可能的结果，并把结果写入 LUT。这样，每输入一个信号进行逻辑运算就等于输入一个地址进行查表，找出地址对应的内容，然后输出存到 RAM 里对应的结果。

如图 1.2.4 所示，一个 LUT 可以随意实现任意 4 输入逻辑功能。

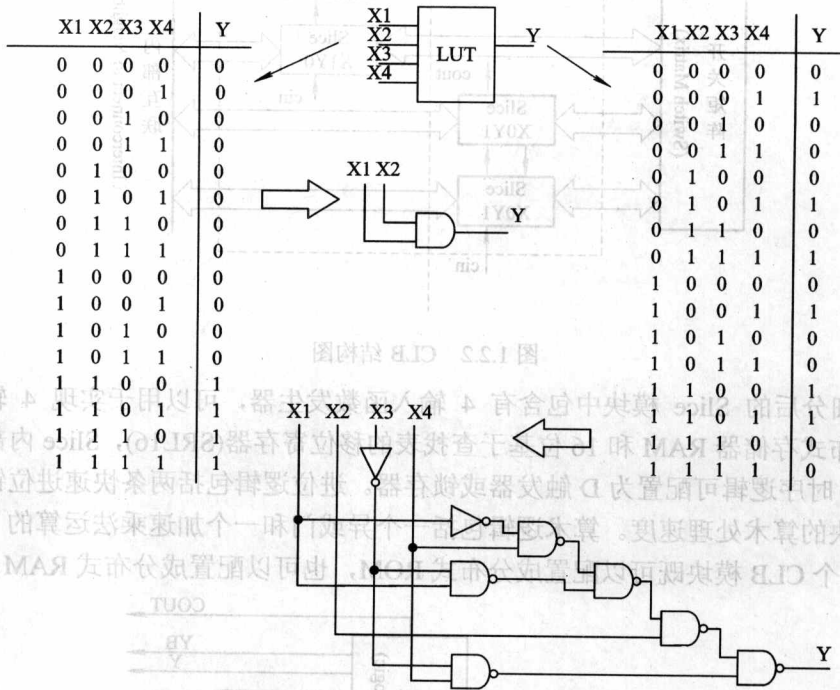


图 1.2.4 LUT 功能实现

如果一个 CLB 要实现 5 输入的逻辑功能，则需要两个 LUT 组合使用，如图 1.2.5 所示，由同一个 CLB 中的两个 LUT 通过一个多路选择器 MUX 组合起来实现 5 输入的逻辑功能。

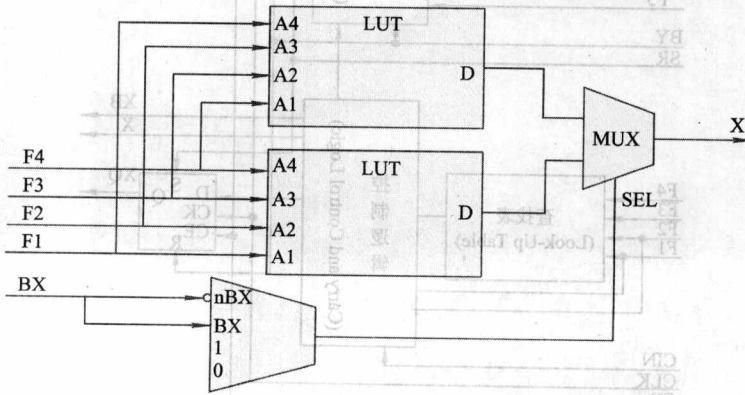


图 1.2.5 两个 LUT 组合实现 5 输入逻辑功能

### 1.2.2 可配置 I/O 模块

可配置 I/O 模块的作用是将外来信号输入到芯片内部，或将信号输出芯片。图 1.2.6 为 Spartan-3E 可配置 I/O 模块的内部结构，主要分三部分：输入通道、输出通道和三态门通道。

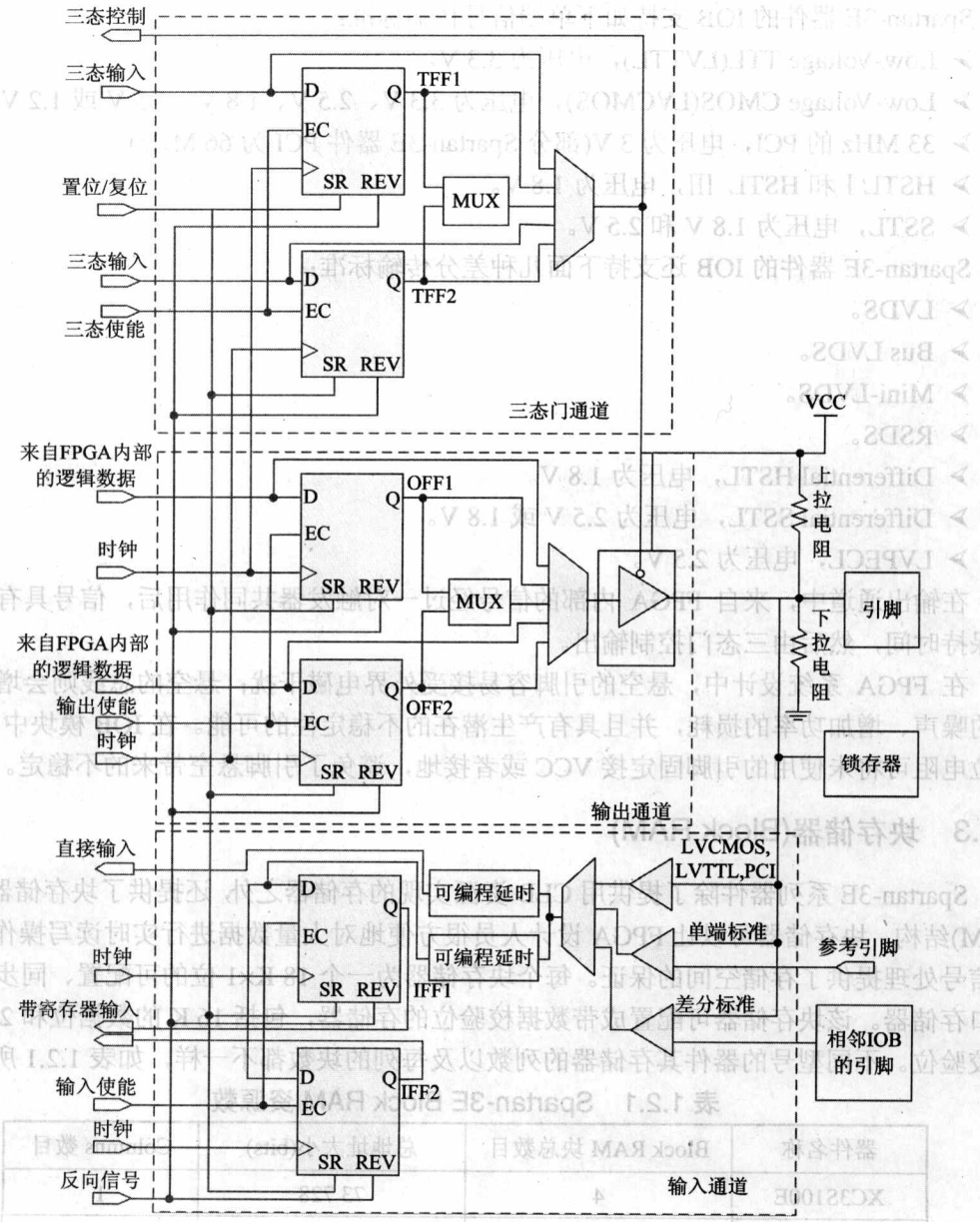


图 1.2.6 可配置 I/O 模块

这三个通道都包含分别由时钟双沿控制的触发器对，因此可以在单个 IOB 上实现 DDR(Double-Data Rate)输入、输出及三态控制。在 DDR 方式设计中，需要提供两个占空比为 50%的时钟，分别提供给一对触发器。DDR 时钟可以将一个时钟反向后生成，也可以通过时钟管理器输出。

在输入通道中，外部输入信号都必须经过一个可编程延时模块，该模块可保证输入信号满足建立时间(Setup Time)和保持时间(Hold Time)的要求。当使用 LVCMOS 和 LVTTTL 标准输入时，Spartan-3E 器件可以设置 2 mA、4 mA、6 mA、8 mA、12 mA、16 mA 和 24 mA 七种输出电流，为信号传输提供了多种电平选择，从而改善板级信号的传输质量。

Spartan-3E 器件的 IOB 支持如下单端信号传输标准：

- Low-Voltage TTL(LVTTTL)，电压为 3.3 V。
- Low-Voltage CMOS(LVCMOS)，电压为 3.3 V、2.5 V、1.8 V、1.5 V 或 1.2 V。
- 33 MHz 的 PCI，电压为 3 V(部分 Spartan-3E 器件 PCI 为 66 MHz)。
- HSTL I 和 HSTL III，电压为 1.8 V。
- SSTL，电压为 1.8 V 和 2.5 V。

Spartan-3E 器件的 IOB 还支持下面几种差分传输标准：

- LVDS。
- Bus LVDS。
- Mini-LVDS。
- RSDS。
- Differential HSTL，电压为 1.8 V。
- Differential SSTL，电压为 2.5 V 或 1.8 V。
- LVPECL，电压为 2.5 V。

在输出通道中，来自 FPGA 内部的信号经过一对触发器共同作用后，信号具有更充裕的保持时间，然后由三态门控制输出。

在 FPGA 系统设计中，悬空的引脚容易接受外界电磁干扰；悬空的总线则会增加系统内的噪声、增加功率的损耗，并且具有产生潜在的不稳定性的可能。在 IOB 模块中，上拉/下拉电阻可将未使用的引脚固定接 VCC 或者接地，避免了引脚悬空带来的不稳定。

### 1.2.3 块存储器(Block RAM)

Spartan-3E 系列器件除了提供用 CLB 资源实现的存储器之外，还提供了块存储器(Block RAM)结构。块存储器可以让 FPGA 设计人员很方便地对大量数据进行实时读写操作，为数字信号处理提供了存储空间的保证。每个块存储器为一个 18 K×1 位的可配置、同步、全双端口存储器。该块存储器可配置成带数据校验位的存储器，包括 16 K 的数据位和 2 K 的奇偶校验位。不同型号的器件其存储器的列数以及每列的块数都不一样，如表 1.2.1 所示。

表 1.2.1 Spartan-3E Block RAM 资源数

| 器件名称      | Block RAM 块总数目 | 总地址大小(bits) | Columns 数目 |
|-----------|----------------|-------------|------------|
| XC3S100E  | 4              | 73 728      | 1          |
| XC3S250E  | 12             | 221 184     | 2          |
| XC3S500E  | 20             | 368 640     | 2          |
| XC3S1200E | 28             | 516 096     | 2          |
| XC3S1600E | 36             | 663 552     | 2          |

Spartan-3E 系列器件的块存储器为完全的双口存储器结构，支持多种数据流操作，可通

过参数设置进行配置。块存储器还支持单、双端口两种模式配置，两种模式下端口定义以及读写方式均有所不同。下面分别对单、双口模式下的端口定义以及读写方式进行详细的说明。

双端口模式如图 1.2.7 所示。

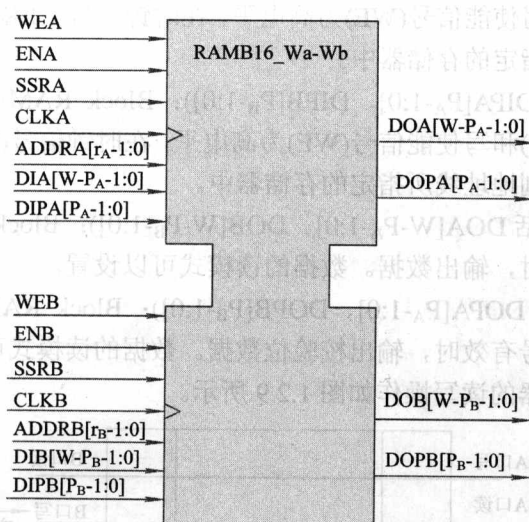


图 1.2.7 双端口模式

单端口模式如图 1.2.8 所示。

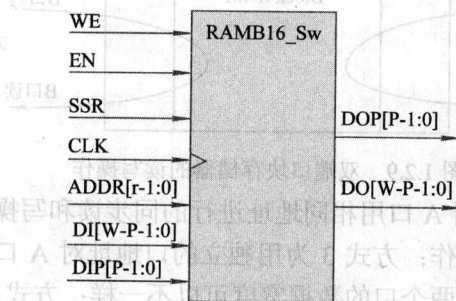


图 1.2.8 单端口模式

两种模式下的端口定义如下：

➤ WE(包括 WEA、WEB)：Block RAM 读/写控制信号。当 EN(ENA、ENB)信号为高时，WE(WEA、WEB)=1 表示对目标地址进行写操作，WE(WEA、WEB)=0 表示对目标地址进行读操作，其读数据的方式由 WRITE\_MODE 来设置。

➤ EN(包括 ENA、ENB)：Block RAM 使能信号。当该控制脚为低时，写入和读出操作无效，DO 和 DOP 保持原有状态和数据。

➤ SSR(包括 SSRA、SSRB)：Block RAM 的同步置位和复位控制信号，当 SSR(包括 SSRA、SSRB)和 EN(包括 ENA、ENB)控制信号为高时，其 DO 和 DOP 的输出为“1”或“0”，这取决于参数 SRVAL。该控制信号不会影响存储器内容和其他端口的操作。

➤ CLK(包括 CLKA、CLKB)：Block RAM 的 A 口/B 口时钟控制信号。时钟的极性可



以选择,既可以用时钟的上升沿,也可以用时钟的下降沿。

➤  $ADDR[r-1:0]$ (包括  $ADDR_A[r_A-1:0]$ 、 $ADDR_B[r_B-1:0]$ ): Block RAM 地址输入信号。用于读/写块存储器内部数据。地址线的宽度与配置的数据位数有关。

➤  $DI[W-P-1:0]$ (包括  $DI_A[W-P_A-1:0]$ 、 $DI_B[W-P_B-1:0]$ ): Block RAM 的数据输入。如果时钟的使能信号(EN)和写使能信号(WE)为高电平,在时钟信号(CLK)上升沿时,将该输入端的数据写入到地址线所指定的存储器中。

➤  $DIP[P-1:0]$ (包括  $DIP_A[P_A-1:0]$ 、 $DIP_B[P_B-1:0]$ ): Block RAM 输入数据的校验信号。如果时钟的使能信号(EN)和写使能信号(WE)为高电平,在时钟信号(CLK)上升沿时,将该输入端的校验位数据写入到地址线所指定的存储器中。

➤  $DO[W-P-1:0]$ (包括  $DO_A[W-P_A-1:0]$ 、 $DO_B[W-P_B-1:0]$ ): Block RAM 的数据输出。在读操作和时钟信号有效时,输出数据。数据的读模式可以设置。

➤  $DOP[P-1:0]$ (包括  $DOP_A[P_A-1:0]$ 、 $DOP_B[P_B-1:0]$ ): Block RAM 输出数据的校验位信号。在读操作和时钟信号有效时,输出校验位数据。数据的读模式可以设置。

双口模式下块存储器的读写操作如图 1.2.9 所示。

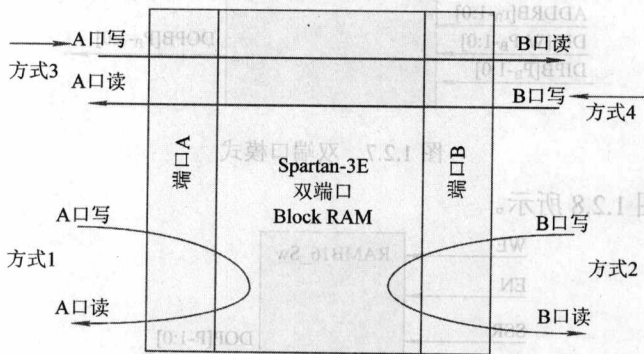


图 1.2.9 双端口块存储器的读写操作

在图 1.2.9 中,方式 1 为对 A 口用相同地址进行的同步读和写操作;方式 2 为对 B 口用相同地址进行的同步读和写操作;方式 3 为用独立的口地址对 A 口进行的写操作, B 口用另一个地址进行读操作,并且两个口的数据宽度可以不一样;方式 4 与方式 3 操作相同,只是数据流方向不同。

在使用 Core Generator(核生成工具)或对源代码例化时,需要对块存储器的一些参数进行设置。其中比较重要的参数设置如下:

➤ **GSR**: 全局的置位和复位控制信号。该信号是 FPGA 器件的控制信号,块存储器本身没有这个控制端。当该信号有效时,块存储器的输出为初始(INIT)值。该信号不会改变存储器的内容。

➤ **INIT\_xx**、**INITP\_xx**: **INIT\_xx** 为数据的初始化值, **INITP\_xx** 为校验位的初始值。

在同时读写 Block RAM 的同一地址时, **WRITE\_MODE** 有三种处理模式。在设计时,可以用 ISE8.2 设计工具的模块设计、约束设计和 Core Generator 来设置。

➤ **WRITE\_FIRST**(写优先)模式:也称为透明模式,为默认状态。在 **WRITE\_FIRST** 模式下,同时读写 Block RAM 同一地址时,读出的数据与写入的数据相同,如图 1.2.10 所示。