普通高等教育“十二五”电子信息类规划教材
卓越工程师教育培养系列教材

数字电路的FPGA设计与实现（基础篇）

刘岚　许建霞　周鹏　黄秋元　陈适　编著

附光盘

机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”电子信息类规划教材
卓越工程师教育培养系列教材

数字电路的FPGA设计与实现
（基础篇）

刘 岚 许建霞 周 鹏 黄秋元 陈 适 编著

机械工业出版社

FPGA 是数字电路或系统设计的通用技术，利用 FPGA 芯片可以实现数字电路的各种功能。本书较系统地介绍了基础数字电路的 FPGA 设计与实现过程。全书共 6 章，主要内容包括：数字电路与 FPGA 技术概述，数字电路基础单元的 FPGA 实现，运算电路的 FPGA 实现，计数器的 FPGA 实现，存储器的 FPGA 实现，接口电路的 FPGA 实现。本书提供了较为丰富的 FPGA 实验例程和设计例程，使读者通过实验和设计更加深入地了解基础数字电路的工作原理，并且逐步掌握 FPGA 的设计与应用技术。

本书中的所有例程均是在 Xilinx11.1 版本仿真环境下进行的，本书所附的光盘为学习者提供了相应的程序和工程文件。

本书深入浅出，实例丰富，取材新颖，图文并茂，叙述详尽清晰，可作为电子信息类本科生和硕士研究生学习 FPGA 应用技术的教材，也可供从事电子电路系统设计的工程技术人员学习参考。

图书在版编目（CIP）数据

数字电路的 FPGA 设计与实现. 基础篇/刘岚编著.
—北京：机械工业出版社，2013.8
普通高等教育“十二五”电子信息类规划教材
ISBN 978-7-111-43524-2

Ⅰ.①数… Ⅱ.①刘… Ⅲ.①数字电路-可编程序逻辑阵列-高等学校-教材 Ⅳ.①TN790.2

中国版本图书馆 CIP 数据核字（2013）第 177161 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：徐 凡 责任编辑：徐 凡
版式设计：霍永明 责任校对：陈秀丽
责任印制：刘 岚
北京京丰印刷厂印刷
2015 年 1 月第 1 版 · 第 1 次印刷
184mm×260mm · 12.25 印张 · 292 字
标准书号：ISBN 978-7-111-43524-2
ISBN 978-7-89405-519-4（光盘）
定价：29.00 元（含 1CD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销售一部：（010）68326294
销售二部：（010）88379649
读者购书热线：（010）88379203

网络服务
教材网：http://www.cmpedu.com
机工官网：http://www.cmpbook.com
机工官博：http://weibo.com/cmp1952

序

自 1985 年 Xilinx 公司推出第一片现场可编程门阵列件（FPGA）至今，FPGA 已经历了二十多年的发展历史。在这二十多年的发展过程中，以 FPGA 为代表的数字系统现场集成技术取得了惊人的发展。

一项重要的工程技术的发展必将会推动工程教育的变革又终将为工程教育所推动。正是由于这样的原因，Xilinx 公司在创办初期就开始了 Xilinx 大学计划（XUP）的推行，其目的是试图促进采用 Xilinx 可编程逻辑器件（PLD）技术的教育和研究。

目前，Xilinx 的 FPGA 器件以及新推出的 ALL Programmable 平台被广泛应用于许多教学领域中，包括数字设计、VHDL 或 Verilog HDL 设计、处理器结构、DSP、电信、可重配置计算机、ASIC 设计及其他可以想到的领域。这表明 Xilinx 可编程逻辑方案已成为许多电子信息和计算机科学系的基本设计课程的一部分。

Xilinx 公司在中国的大学计划正在蓬勃地开展着，目前已在清华大学、北京大学、西安电子科技大学、电子科技大学、中国科学技术大学、上海交通大学、北京理工大学、西安交通大学、复旦大学、武汉大学、华中科技大学、武汉理工大学等近百所高校建立了 Xilinx 联合实验室，希望通过 Xilinx 大学计划，帮助国内的广大师生提高 FPGA 的教学和科研水平。

武汉理工大学 Xilinx 联合实验室的老师和学生们将自己学习和研究成果梳理汇聚，撰写出了这本既面向高校教学使用同时又适用于科技人员参考的《数字电路的 FPGA 设计与实现（基础篇）》。该书所介绍的一系列开发应用环境以及基础性实验将有助于初学者尽快熟悉 FPGA 的应用流程，同时书中所描述的设计实例又将为广大读者提供解决实际问题的思路和方法，相信此书必将为 FPGA 应用技术的教学和相关科研工作的发展起到一定的促进和推动作用。

Xilinx 中国大学计划负责人　谢凯年

前 言

近年来，在数字电路的设计方面，可编程逻辑器件（PLD）被广泛采用。

早期的可编程逻辑器件，如简单可编程逻辑器件（SPLD）、可编程逻辑阵列（PLA）和通用阵列逻辑（GAL）等因为结构简单，只能实现较小规模的电路。为了完成复杂的设计，各类高密度的可编程逻辑器件相继推出，其中以 Xilinx 公司的 FPGA 和 Altera 公司的复杂可编程逻辑器件（CPLD）最为出众。

以硬件描述语言（VerilogHDL 或 VHDL）所完成的电路设计，可以经过简单的综合与布局，快速地烧录至 FPGA 上进行测试，这已成为现代 IC 设计验证的主流技术。

目前，高校中强调理论教学与实践相结合的呼声越来越高，学生在掌握基本理论知识的基础上，对提高实践能力的需求也越来越强烈。近年来，教育部开始在一些高校启动了“卓越工程师教育培养计划”，其宗旨就是要从教学体系上和教学内容上强化学生的综合能力。本书是卓越工程师教育培养系列教材中的一本，该系列教材包括：《FPGA 应用技术基础教程》《数字电路的 FPGA 设计与实现（基础篇）》《数字电路的 FPGA 设计与实现（应用篇）》《通信系统中编解码技术的 FPGA 设计与实现》《通信系统中同步技术的 FPGA 设计与实现》《通信系统中调制解调技术的 FPGA 设计与实现》《数字信号处理系统的 FPGA 设计与实现》等。

本书的主要特色：结合 Xilinx 公司的 FPGA 系统，通过一系列的实验环节，介绍了基础数字电路的 FPGA 应用设计、仿真与实现的技术流程。本书注重动手能力的培养，重视具体设计与实现，书中的所有过程都体现了由基础理论到应用设计的描述，从而引导读者由浅入深地实现从数字电路基本功能的设计到微处理器控制下的应用设计。特别是本教程通过大量的简单设计实例。将 FPGA 与微处理器的嵌入式设计引入到学习过程中，强化了读者对片上系统的理解和应用。

本书是在 Xilinx 公司推广的中国大学计划的支持下，在其组织者谢凯年博士的鼓励下编著的。武汉理工大学与 Xilinx 公司共同组建的信号传输与处理联合实验室强化了我们学习、研究和应用 FPGA 技术的团队，本书所依据的素材和工程文件均由这个团队的师生所验证，本书是他们辛勤劳动的果实。

本书可作为电子信息类本科生和硕士研究生学习 FPGA 应用技术的教材，也可供从事电子电路系统设计的工程技术人员学习参考。希望读者通过本教程的学习能够较快地了解和掌握 FPGA 的基本知识和应用技术以及 FPGA 的设计思想和设计方法。

本书承蒙华中科技大学杨晓非教授和武汉理工大学吴友宇教授审阅，他们对本书的编写提出了不少宝贵意见和有益的建议，在此表示诚挚的感谢。

由于编著者的水平和经验有限，书中难免存在错误和不妥之处，敬请广大读者批评指正。

编著者

目　录

第 1 章　数字电路与 FPGA 技术概述

1.1　数字电路概述

1.1.1　数字电路的特点

数字化是进入信息化时代的必要条件，所谓数字化是将信息用数字 0、1 编码来表述。0 和 1 两个数码在电路中表示两种对立状态，如电压的高低、电流的大小、脉冲的有无等，基于这种原理的电路即是数字电路。

数字电路主要有以下优点：

1）电路可稳定工作在不连续的、特征差别大的两个对立状态下，因此电路的可靠性和稳定性非常高。

2）从理论上讲，数字信号表述的信息在处理过程中不会产生失真。

3）信息的传输、运算、处理和保存变得更为方便。

4）对电路的实时控制准确有效。

5）与计算机及外围电路兼容，便于运用计算机进行运算和控制处理。

6）数字电路可以很方便地级连和扩展，形成中大规模数字集成电路。

基于上述原因，数字电路已被广泛应用于工业、交通、军事和广播等几乎所有领域，在消费电子方面的发展也方兴未艾。

现代数字电路由若干数字集成器件构造而成，数字集成器件所用的材料以硅材料为主。在高速电路中，也使用化合物半导体材料，如砷化镓等。逻辑门是数字电路中一种重要的逻辑单元电路，TTL 逻辑门电路问世较早，其工艺经过不断改进，至今仍为主要的基本逻辑器件之一。随着 CMOS 工艺的发展，TTL 的主导地位受到了动摇，有被 CMOS 器件取代的趋势。近年来，PLD 特别是 FPGA 的飞速进步，使数字电子技术开创了崭新的局面。这些器件不仅规模大，而且具有硬件与软件相结合的特点，器件的功能更加完善，使用更加灵活。

1.1.2　数字电路的分类

按功能分类，数字电路主要有组合逻辑电路和时序逻辑电路。

1. 组合逻辑电路

组合逻辑电路由各种门电路组成，其特点是某一时刻的输出状态取决于该时刻电路的输入状态的组合关系。常用的组合逻辑电路有加法器、译码器、数据选择器等。组合逻辑电路的功能一般用逻辑图、逻辑表达式、真值表等形式进行描述。在进行逻辑电路的分析和实际电路设计时，通常需列出真值表，分析其逻辑关系，得到逻辑命题的函数表达式，而后实现逻辑命题的逻辑电路。图 1-1 是 4 选 1 数据选择器的电路图，电路中包含了基本的与门、非

门、异或门。

2. 时序逻辑电路

时序逻辑电路除了用各种逻辑门电路组成组合电路外，还用具有存储功能的触发器构成存储电路，用于实现某种具有时序逻辑功能的复杂逻辑电路。它与组合逻辑电路的区别是，电路的输出状态不仅由电路的输入信号状态组合关系决定，同时还与该电路的现态有关。这就需要电路中包含具有记忆功能的元件和反馈通道，以便使记忆下来的状态能在下一时刻影响电路。图1-2所示为带使能端的RS锁存器的电路图，图中有时钟控制和反馈电路，可以记忆前一时刻的输出。

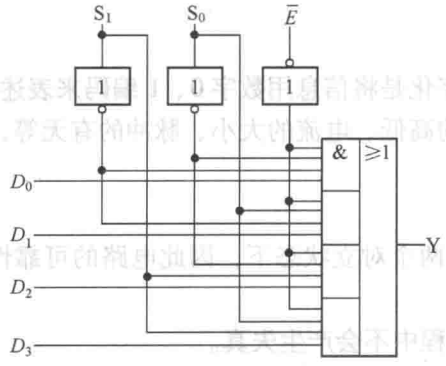

图1-1　4选1数据选择器电路

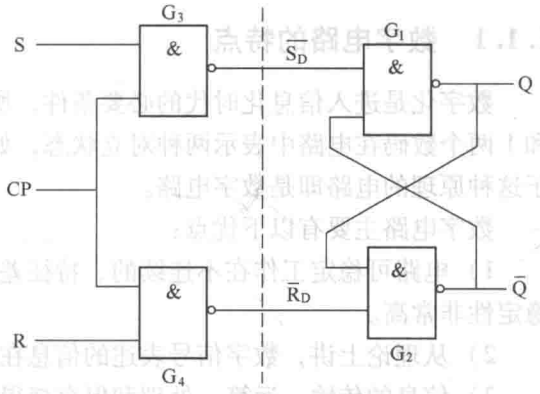

图1-2　带使能端的RS锁存器

时序逻辑电路根据触发器动作方式的不同可分为同步时序逻辑电路和异步时序逻辑电路。同步时序逻辑电路中各个触发器都统一在一个时钟脉冲下工作；异步时序逻辑电路中各个触发器可以在不同的时钟脉冲下工作。因而，同步时序逻辑电路里触发器状态的改变是同时的，而异步时序逻辑电路里触发器状态的改变是有先后顺序的。在分析和设计异步时序逻辑电路时，需要分析电路状态的变化规律，以及在输入信号和时钟信号下的输出情况。

对于异步时序逻辑电路的描述，一般有4种方法：逻辑方程、状态转换表、状态转换图和时序图。在分析和设计时序逻辑电路时，通过逻辑方程将各种输入和电路状态的组合带入输出方程和状态方程，得到反映电路的状态转换和输出变化情况的逻辑状态转换图，从而实现时序电路的逻辑功能。

1.1.3　数字逻辑电路设计中的重要问题

数字逻辑电路设计中的两个重要问题是电路的简化与电路性能指标的提高。电路简化的目的是要用最少的元器件来实现电路，以降低电路的成本和出现错误的概率；电路性能指标包括电路的速度、电路的扇入、扇出系数和可靠性等。这两方面的问题既统一又矛盾，在电路设计过程中必须要综合考虑。

1. 化简数字逻辑电路

一般来说，数字逻辑电路是根据逻辑表达式得到的。逻辑表达式的形式多种多样，因此简化电路也相应地存在着多种多样的形式，应视具体情况进行选择、评价。对完全确定的单输出逻辑电路，可按照实际要求选择适当的简化方法（如代数法、卡诺图法等），使逻辑表

达式达到最简即可；对于带任意项的逻辑问题必须充分考虑其任意项，这样才能简化逻辑电路。多输出电路设计时要充分考虑各输出函数的公用项，找出其共享部分，将各个函数化为最简。对无反变量输出逻辑电路设计，直接的办法是当需要某个反变量时就用非门将相应的原变量转换成反变量，但这样处理往往不经济。

2. 提高数字逻辑电路的性能指标

在数字逻辑电路设计过程中，不仅要使电路最简单，性能指标的提高也很重要。当电路的级数增多时，输出相对于输入的传输延时就会增大，从而造成电路的工作速度不能满足要求；当门电路的扇入、扇出系数超出现有组件的技术指标时，又需要采用增加级数即降低速度来减少扇入、扇出系数的要求。一般情况下，提高数字逻辑电路性能的方法有3种。

1）压缩级数提高电路速度。电路的级数反映在输出函数表达式中，即“与”、“或”、“非”运算的层次数。因此，压缩级数可通过对输出函数求反或展开来获得。

2）增加级数满足门电路的扇入、扇出系数的要求。在有些情况下，当输入变量的数目增多时，门电路的扇入、扇出系数受到限制。这时必须增加电路的级数，降低每级输入变量的个数从而达到降低门电路的扇入、扇出系数的要求。因此，通常采用矩阵结构和树型结构来实现。

3）消除竞争冒险，提高电路的可靠性。通常情况下，所设计的电路的输入、输出之间的关系是电路在稳定状态下的逻辑关系，并未考虑输入信号传递中经历的过渡过程。但实际过程中信号的变化不是即时的，存在一定的边沿时间，即信号在电路中传递会有导线的传播时间和通过门时的延迟时间。因此，在输入信号变化后的短暂时间内，输入、输出之间的关系不一定符合逻辑表达式所描述的逻辑关系，即输出电路中可能存在尖峰干扰脉冲现象。这种干扰脉冲可能会造成电路的误动作，尤其在时序电路中影响可能会更大。在这种情况下，必须消除电路中的险象。消除冒险现象的主要方法：增加多余项或以多余因子消除逻辑冒险现象，在输出端连接低通环节以减轻干扰，利用取样脉冲避开冒险现象。

1.2 C语言与Verilog HDL的区别与联系

对数字电路的描述可以选择C语言和Verilog HDL语言。C语言通常用来做算法的描述和验证，但其数据处理速度与程序的大小和计算机的运行速度有关，不能独立于计算机而存在。Verilog HDL语言是硬件描述语言，可使用Verilog HDL编写硬件描述语言程序进行数字电路设计。使用Verilog HDL程序设计硬件电路的好处是易于理解、易于维护、调试电路速度快，但这样设计所得到的电路还需进行仿真，以便从电路结构上保证算法能在规定的时间内完成，并能与前端和后端的设备或器件正确无误地交换数据。这就需要用C语言和Verilog HDL配合，进行逻辑设计仿真，以验证电路功能的正确性。

1.2.1 C语言与Verilog HDL的区别

C语言是目前世界上应用最为广泛的一种编程语言，有可靠的编程环境，语言完备，缺陷较少。此外，C语言灵活且功能强大，还可以通过编程语言接口（PLI）编写自己的系统任务直接与硬件仿真器结合使用。

相比来看，Verilog HDL只是针对硬件进行描述的语言，在其他环境下使用并不方便。

而且 Verilog HDL 的仿真、综合和差错工具等大部分软件都是商业软件，与 C 语言相比缺乏长期大量的使用，也有很多缺陷。

在运行过程中，C 语言是按行依次执行的，属于顺序结构。而 Verilog HDL 描述的硬件是可以在同一时间同时运行，属于并行结构。

1）C 程序调用函数没有延时特性，对同一个函数的不同调用是一样的。而 Verilog HDL 中对模块的不同调用却是不同的，即使调用的是同一个模块，也必须用不同的名字来指定。

2）C 语言的变化很多，转换过程会遇到一些困难。Verilog HDL 的语法规则较为死板，限制很多，能用的判断语句有限，仿真速度较慢，差错功能差，错误信息不完整。

3）C 语言没有时间关系，而 Verilog 程序则必须做到没有任何外加的人工延时信号，否则将无法使用综合工具把 Verilog 源代码转化为门级逻辑。

1.2.2　C 语言与 Verilog HDL 的联系

C 语言与 Verilog HDL 有很多共通性，在一些关键字和运算符上，可以看到两者的联系，如表 1-1 和表 1-2 所示。

表 1-1　C 语言与 Verilog HDL 相对应的关键字与控制结构

C	Verilog	C	Verilog
sub-function	module、function、task	while	while
if-then-else	if-then-else	break	disable
case	case	define	define
{,}	begin、end	int	int
for	for	printf	monitor、display、strobe

表 1-2　C 语言与 Verilog HDL 相对应的运算符

C	Verilog HDL	功　能	C	Verilog HDL	功　能
*	*	乘	!	!	反逻辑
/	/	除	&&	&&	逻辑与
+	+	加	‖	‖	逻辑或
-	-	减	>	>	大于
%	%	取模	<	<	小于

综上所述，C 语言与 Verilog HDL 的联系如下：

1）C 语言与 Verilog HDL 可以配合使用，辅助设计硬件。

2）C 语言与 Verilog HDL 很相像，但要稍加限制。

3）C 语言程序很容易转化成 Verilog 程序。

1.3　FPGA 基本结构介绍

在数字电路设计中，PLD 被广泛采用。根据所有电路都是由门电路组成这一思想，PLD 内部的电路设计也可由 FPGA 通过连线构成，这就是 FPGA 设计的原理。FPGA 器件一般都

是由查找表组成的基本逻辑单元，输入/输出（I/O）单元以及连线通道3部分组成。当然，基本逻辑单元已经从最初的简单门单元发展到可以使用选择器、触发器和缓冲器，甚至是SRAM等功能更强的模块。不过，采用SRAM工艺制造的模块在使用时必须外加EEPROM，以便存储下载文件。FPGA的基本结构如图1-3所示。

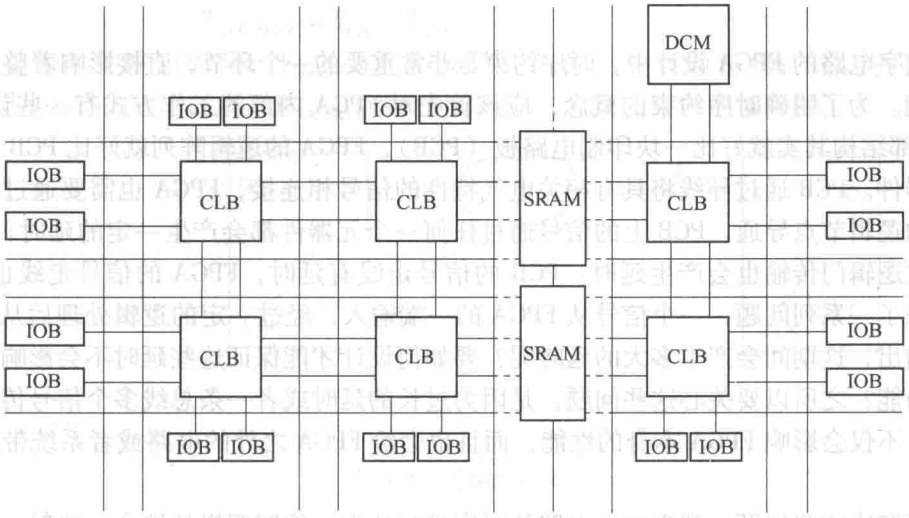

图1-3　FPGA基本结构

图中各模块说明如下：

1）CLB是FPGA内的基本逻辑单元。CLB的实际数量和特性会依器件的不同而不同，但是每个CLB都包含一个可配置开关矩阵，此矩阵由4个或6个输入、一些选型电路（多路复用器等）和触发器组成。开关矩阵是高度灵活的，可以对其进行配置以便处理组合逻辑、移位寄存器或RAM。每个CLB模块不仅可以用于实现组合逻辑、时序逻辑，还可以配置为分布式RAM和分布式ROM。

2）可编程I/O单元IOB是芯片与外接电路的接口部分，完成不同电气特性下对I/O信号的驱动与匹配要求。通过软件的灵活配置，可适配不同的电气标准与I/O物理特性，即可以调整驱动电流的大小，可以改变上、下拉电阻。目前，I/O口的频率越来越高，一些高端的FPGA通过DDR寄存器技术可以支持高达2Gbit/s的数据传输速率。

3）DCM是数字时钟管理模块，提供数字时钟管理和相位环路锁定功能。相位环路锁定能够提供精确的时钟综合，且能够降低抖动，并实现过滤功能。

4）嵌入式模块RAM（BRAM）可被配置为单端口RAM、双端口RAM、内容地址存储器（CAM）以及FIFO（等常用存储结构。单片模块）RAM的容量为18KB，即位宽为18bit、深度为1024，可以根据需要改变其位宽和深度，但要满足两个原则：首先，修改后的容量（位宽深度）不能大于18KB；其次，位宽最大不能超过36bit。

5）布线资源可连通FPGA内部的所有单元，而连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。FPGA芯片内部有着丰富的布线资源，根据工艺、长度、宽度和分布位置的不同而划分为4种不同的类别。第一类是全局布线资源，用于芯片内部全局时钟和全局复位/置位的布线；第二类是长线资源，用以完成芯片Bank间的高速信号和第二全局时

钟信号的布线；第三类是短线资源，用于完成基本逻辑单元之间的逻辑互连和布线；第四类是分布式的布线资源，用于专有时钟、复位等控制信号线。

1.4 时序分析和约束条件

在数字电路的FPGA设计中，时序约束是非常重要的一个环节，直接影响着整个电路的工作性能。为了明确时序约束的概念，应该首先对FPGA内部的工作方式有一些认识。FPGA的内部结构其实就好比一块印制电路板（PCB），FPGA的逻辑阵列就好比PCB上的一些分立元器件。PCB通过导线将具有相关电气特性的信号相连接，FPGA也需要通过内部连线将相关的逻辑节点导通。PCB上的信号通过任何一个元器件都会产生一定的延时，FPGA的信号通过逻辑门传输也会产生延时。PCB的信号走线有延时，FPGA的信号走线也有延时。这就带来了一系列问题，一个信号从FPGA的一端输入，经过一定的逻辑处理后从FPGA的另一端输出，这期间会产生多大的延时呢？要如何设计才能保证这些延时不会影响电路系统的逻辑功能？之所以要关心这些问题，是因为过长的延时或者一条总线多个信号传输时间的不一致，不仅会影响FPGA本身的性能，而且也会给FPGA之外的电路或者系统带来诸多问题。

为了解决这些问题，就需要对电路的时序进行约束，控制逻辑的综合、映射、布局和布线，以减小逻辑和布线延时，从而提高工作频率，使系统工作在可靠的状态下。

时序约束主要包括周期约束（FFS到FFS，即触发器到触发器）、偏移约束（IPAD到FFS、FFS到OPAD）以及静态路径约束（IPAD到OPAD）。通过附加约束条件可以使综合布线工具调整映射和布局布线过程，使设计达到时序要求。例如，使用OFFSET _ IN _ BEFORE约束可以告诉综合布线工具输入信号在时钟之前什么时候准备好，综合布线工具就可以根据这个约束调整与IPAD相连的Logic Circuitry的综合实现过程，使结果满足FFS的建立时间要求。

附加时序约束的一般策略是先附加全局约束，然后对快速和慢速例外路径附加专门约束。附加全局约束时，首先定义设计的所有时钟，对各时钟域内的同步元件进行分组，对分组附加周期约束，然后对FPGA/CPLD输入与输出PAD附加偏移约束，并对全组合逻辑的PAD到PAD路径附加约束。附加专门约束时，首先约束分组之间的路径，然后约束快、慢速例外路径和多周期路径，以及其他特殊路径。

1.4.1 周期约束

1. 周期约束概念

周期约束是一个基本时序和综合的约束，它附加在时钟网线上，时序分析工具根据PERIOD约束检查时钟域内所有同步元件的时序是否满足要求。PERIOD约束会自动处理寄存器时钟端的反相问题，如果相邻同步元件时钟相位相反，那么它们之间的延迟将被默认限制为PERIOD约束值的1/2。

周期的大小是如何确定的呢？实际上，硬件设计电路所能工作的最高频率取决于芯片内部元件本身固有的建立保持时间，以及同步元件之间的逻辑和布线延迟。所以，电路最高频率由代码和芯片两部分共同决定。相同的程序，在速度等级高的芯片上能达到更高的最高工

作频率。同样，在同一芯片内，经过速度优化的代码具有更高的工作频率，在实际中往往取二者的平衡。

一般情况下，周期的算术定义如图 1-4 所示。时钟的最小周期为

$$T_{CLK} = T_{CKO} + T_{LOGIC} + T_{NET} + T_{SETUP} - T_{CLK\_SKEW}$$

$$T_{CLK\_SKEW} = T_{CD2} - T_{CD1}$$

其中，T_{CKO}为时钟输出时间；T_{LOGIC}为同步元件之间的组合逻辑延迟；T_{NET}为网线延迟；T_{SETUP}为同步元件的建立时间；T_{CLK_SKEW}为时钟信号 T_{CD2}和 T_{CD1}时钟的偏移量。

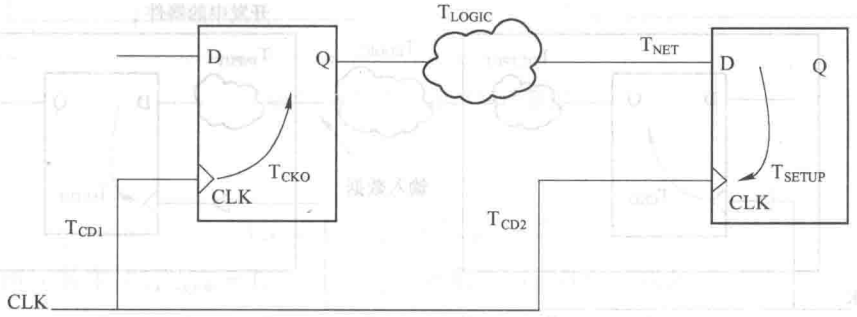

图 1-4　周期的定义

2. 周期约束方法

在添加时钟周期的约束之前，需要对电路的期望时钟周期有一个合理的估计，这样才不会附加过松或过紧的周期约束。过松的约束往往不能达到性能要求，过紧的则约束会增加布局布线的难度，实现的结果也不一定理想。常用的工程策略：附加的时钟周期约束的时长为期望值的 90%，即约束的最高频率是实际工作频率的 110% 左右。

附加时钟周期约束一般是直接将周期约束附加到寄存器时钟网线上，其语法为

［约束信号］PERIOD = {周期长度} {HIGH | LOW}［脉冲持续时间］　(1-1)

其中，[ ] 内的内容为可选项；{ } 中的内容为必选项；“ | ”表示选择项；［约束信号］可为“Net net _ name”或“TIMEGRP group _ name”，前者表示周期约束作用到线网所驱动的同步元件上，后者表示约束到 TIMEGRP 所定义的信号分组上（如触发器、锁存器及 RAM 等）；{周期长度} 为要求的时钟周期，可选用 ms、s、ns 及 ps 等单位，默认值为 ns，对单位不区分大小写；{HIGH | LOW} 用于指定周期内第一个脉冲是高电平还是低电平；［脉冲持续时间］用于指定第一个脉冲的持续时间，可选用 ms、s、ns 以及 ps 等单位，默认值为 ns，如果默认该项，则默认为 50% 的占空比。如语句

Net “ clk _ 100MHz” period = 10ns High 5ns;

指定了信号“clk _ 100MHz”的周期为 10ns，高电平持续的时间为 5ns，该约束将被添加到信号 clk _ 100MHz 所驱动的元件上。

1.4.2　偏移约束

为了确保芯片数据采样可靠和下级芯片之间正确地交换数据，需要约束外部时钟和数据输入输出引脚之间的时序关系。约束的内容为告诉综合器、布线器输入数据到达的时刻，或者输出数据稳定的时刻，从而保证与下一级电路的时序关系。这种时序约束在 Xilinx 产品中

用 Setup to Clock（edge），Clock（edge）to hold 等表示，在 Altera 产品中则常用 tsu（Input Setup Times）、th（Input Hold Times）、tco（Clock to Out Delays）来表示。很多其他时序工具直接用 setup 和 hold 表示。其实它们所要描述的是同一个问题，仅仅是时间节点的定义上略有不同。下面依次介绍各种时序关系。

1. 偏移约束分类

（1）关于输入到达时间

Xilinx 的“输入到达时间的计算”时序描述如图 1-5 所示。

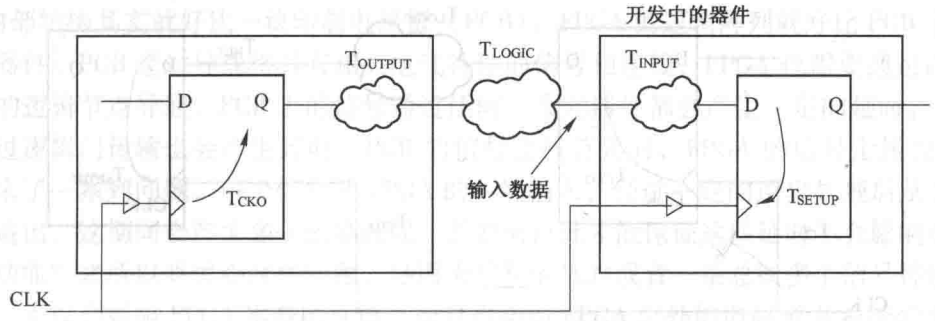

图 1-5　输入到达时间示意图

定义的含义是输入数据在有效时钟沿之后的 $T_{ARRIVAL}$ 时刻到达，即

$$T_{ARRIVAL} = T_{CKO} + T_{OUTPUT} + T_{LOGIC} \tag{1-2}$$

根据式（1-1），可得

$$T_{CKO} + T_{OUTPUT} + T_{LOGIC} + T_{INPUT} + T_{SETUP} - T_{CLK\_SKEW} = T_{CLK}; \tag{1-3}$$

将式（1-2）代入式（1-3），可得

$$T_{ARRIVAL} + T_{INPUT} + T_{SETUP} - T_{CLK\_SKEW} = T_{CLK} \tag{1-4}$$

而 T_{CLK_SKEW} 满足时序关系后为负，所以，$T_{ARRIVAL}$ 应该满足时序关系：

$$T_{ARRIVAL} + T_{INPUT} + T_{SETUP} < T_{CLK} \tag{1-5}$$

其中，T_{INPUT} 为输入端的组合逻辑、网线和 PAD 的延迟之和；T_{SETUP} 为输入同步元件的建立时间。

（2）数据延时和数据到达时间的关系

T_{DELAY} 为要求的芯片内部输入延迟，其最大值 T_{DELAY_MAX} 与输入数据到达时间 $T_{ARRIVAL}$ 的关系如图 1-6 所示，即

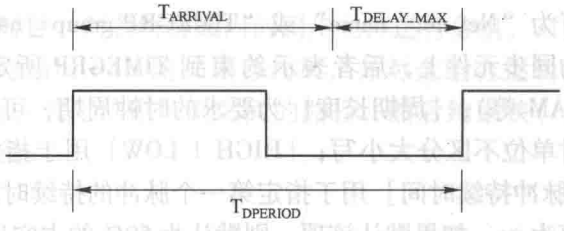

图 1-6　数据延时和数据到达时间示意图

$$T_{DELAY\_MAX} + T_{ARRIVAL} = T_{PERIOD} \tag{1-6}$$

所以

$$T_{DELAY} < T_{DELAY\_MAX} = T_{PERIOD} - T_{ARRIVAL} \tag{1-7}$$

（3）要求输出的稳定时间

从下一级输入端的延迟可以计算出当前设计输出的数据必须在何时稳定下来，根据这个数据对设计输出端的逻辑布线进行约束，以满足下一级的建立时间要求，保证下一级采样的数据是稳定的。计算要求的输出稳定时间的示意图如图 1-7 所示。

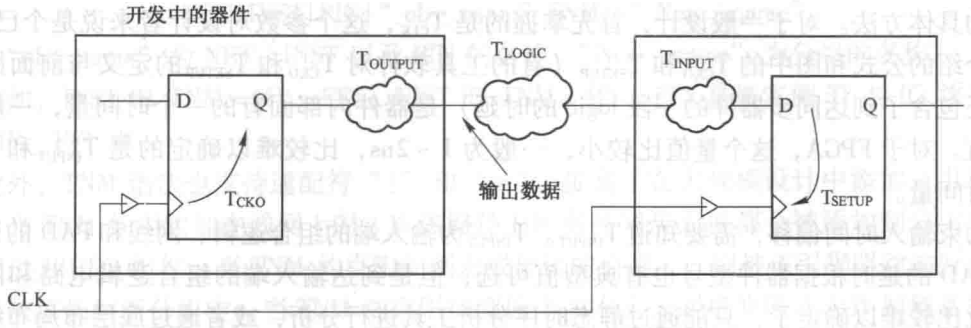

图 1-7　输出稳定时间示意图

定义

$$T_{STABLE} = T_{LOGIC} + T_{INPUT} + T_{SETUP}$$

可以得到（其中 $T_{CLK\_SKEW} = T_{CLK1} - T_{CLK2}$）周期（Period）公式为

$$T_{CLK} = T_{CKO} + T_{OUTPUT} + T_{LOGIC} + T_{INPUT} + T_{SETUP} + T_{CLK\_SKEW}$$

将 T_{STABLE} 的定义式代入到周期公式，可得

$$T_{CLK} = T_{CKO} + T_{OUTPUT} + T_{STABLE} + T_{CLK\_SKEW}$$

所以

$$T_{CKO} + T_{OUTPUT} + T_{STABLE} < T_{CLK}$$

其中，T_{OUTPUT} 为设计中连接同步元件输出端的组合逻辑、网线和 PAD 的延迟之和；T_{CKO} 为同步元件时钟输出时间。

这个公式就是 T_{STABLE} 必须满足的基本时序关系，即本级的输出应该保持怎么样的稳定状态，才能保证下级芯片的采样稳定。有时也称这个约束关系为输出数据的保持时间时序约束关系，只要满足上述关系，当前芯片输出端的数据比时钟上升沿提早 T_{STABLE} 时间稳定下来，下一级就可以正确地采样数据。

2. 偏移约束方法

偏移约束语句只能用于端口信号，不能应用于内部信号，包括 OFFSET _ IN _ BEFORE，OFFSET _ IN _ AFTER，OFFSET _ OUT _ BEFORE，OFFSET _ OUT _ AFTER 等 4 类基本约束语句。

偏移约束的基本语法为

OFFSET = [IN | OUT]“ offset _ time” [units] {BEFORE | AFTER}“ clk _ name”
[TIMEGRP“ group _ name”];

其中，[IN | OUT] 说明约束的是输入还是输出；“offset _ time” 为数据和有效时钟沿之间的时间差；{BEFORE | AFTER} 表明该时间差是在有效时钟之前还是之后；“clk _ name” 为有效时钟的名字；[TIMEGRP “group _ name”] 为用户添加的分组信号，在默认时，默认为时钟 clk _ name 所驱动的所有触发器。

偏移约束通知布局布线器输入数据的到达时刻，从而可准确调整布局布线的过程，使约束信号建立时间满足要求。