

# FPGA 片内延时锁相环架构研究与设计

## 摘 要

FPGA（现场可编程门阵列）芯片是 IC 领域的重要组成部分，越来越多的集成电路设计和仿真依靠其来完成。目前市场上的 FPGA 芯片以欧美大公司的产品占主导，国内的可编程逻辑芯片设计和制造尚处于起步阶段，因此这一方向的研究具有重要的价值和意义。论文以一款国内自主研发的 0.25um CMOS 工艺 SRAM 型 FPGA 芯片设计为基础，主要介绍了 FPGA 芯片内延时锁相环架构研究设计及其应用，并在原有架构的基础上提出并设计了兼具更快的锁定速度和稳定的时钟调节能力的新型延时锁相环架构 OSDLL。

论文介绍了延时锁相环（Delay Locked Loop）的工作原理，电路结构组成，子模块功能定义以及设计等，其中数字控制逻辑依据半定制数字电路设计流程设计完成，鉴相器，可调延时链和时钟生成模块等依据全定制流程设计完成。经过数模混合仿真平台，设计仿真通过并整合至 FPGA 芯片内部。此 100 万门 FPGA 芯片完成流片后，MPW 功能及性能测试达到设计指标。该 FPGA 内 DLL 模块的工作时钟的输入范围为 20MHz 到 200MHz，与同类产品相比具有更高的精度、更低的功耗以及达到了较高的性能指标。

OSDLL 的设计是在原有 DLL 架构的基础上融入了 one-shot 延时计算技术，优化了 DLL 时钟调节原理。在不过多增加芯片面积的情况下较大程度的减少了 DLL 锁定时间，进一步提高了 DLL 和用户设计的系统性能。

DLL 应用领域极其广泛，例如可以作为时钟管理模块嵌入到 ASIC 或 FPGA 芯片内部，也可以作为 IP 在 SOC 系统中成为独立的集成子模块。在 FPGA 内可以完成时钟同步，分频，倍频等时钟管理功能，同时可以作为板级设计提供时钟源，简化设计。

关键词：延时锁相环； FPGA； 快速锁定； 时钟管理技术；

# **Research and Design of FPGA Delay Locked Loop Architecture**

## **ABSTRACT**

FPGA (Field Programmable Gate Array) chip is an important part of IC industry products. More and more IC designs and simulations are implemented using FPGA. Today's FPGA chip market is dominated by US or Europe companies and almost no FPGA series product is designed and manufactured in China, which makes this paper significant valuable according to this research direction. Based on a SRAM FPGA chip project that is fabricated using the SMIC 0.25um CMOS process, research and design of FPGA inside-chip Delay Locked Loop (DLL) architecture is presented and a new fast-lock version DLL architecture, OSDLL, which can balance lock time cost and stable clock management performance, is introduced and designed.

The DLL designed in the thesis is a part of the FPGA chip project. In this paper the principles and implementation of the basic modules are presented and the frame of the whole circuit is introduced, in which digital control logic is designed through half custom flow and other sub-modules such as phase detector, programmable delay chain and clock generator are designed according to full custom design flow. After digital & analog mix-simulation of DLL, the whole design is embedded in FPGA chip circuit and this FPGA is fabricated using the SMIC 0.25um CMOS process. The test result of FPGA MPW shows that DLL runs well both in function and performance perspectives. The input frequency range of the FPGA DLL designed in the thesis can be operated from 20MHz-200MHz. Compared with the counterpart designs, the proposed DLL has the advantages of high precision and low power consumption with high performances.

Based on the formal version DLL, a new Fast-lock Delay-Locked Loop, OSDLL, is designed. OSDLL can lock much faster than DLL while it still maintains stable clock management ability. One-shot delay calculation technology is used in OSDLL and works smoothly together with DLL adjusting mechanism. OSDLL doesn't cost much more silicon space and raises lock speed tremendously which brings the performance of DLL and user design to another level.

DLL module can be used widely in IC design field as embedded clock management circuit in ASIC or FPGA and build-in IP in SOC. In FPGA chip DLL can synchronize clock, act as a clock doubler or divide the user source clock and also provide high quality clock source to board level to simplify design.

**Keywords:** Delay Locked Loop; FPGA; fast lock; clock management;

## 插图清单

图 2-1 DLL 时钟管理举例 .....	6
图 2-2 DLL 与 PLL 简化结构对比图 .....	7
图 2-3 DLL 简化结构框图 .....	8
图 2-4 DLL 时钟调整前波形 .....	9
图 2-5 DLL 时钟调整后波形 .....	9
图 2-6 DLL 模块结构图 .....	10
图 2-7 锁定窗图示 .....	12
图 2-8 时钟移相器基本结构 .....	13
图 2-9 时钟移相器的可配置结构 .....	14
图 2-10 时钟生成模块原理示意图 .....	14
图 2-11 状态机控制器原理示意图 .....	15
图 2-12 one-shot 原理结构示意图 .....	16
图 2-13 one-shot 原理波形示意图 .....	16
图 2-14 OSDLL 时钟调整过程 .....	18
图 3-1 FPGA 芯片中 DLL 结构框图 .....	19
图 3-2 DLL 数字控制逻辑和可调延时链 .....	20
图 3-3 DLL 工作节拍流程 .....	21
图 3-4 DLL 状态跳转与时钟调整 .....	22
图 3-5 状态机状态转移图 .....	23
图 3-6 锁定后 COUNTER_JF 调整 .....	23
图 3-7 JF 逻辑结构示意图 .....	24
图 3-8 数字控制部分综合后电路框图 .....	25
图 3-9 鉴相器结构框图 .....	25
图 3-10 鉴相器电路结构图 .....	26
图 3-11 鉴相器超前/滞后信号(CLK_FB 滞后于 CLK_REF) .....	27
图 3-12 鉴相器超前/滞后信号(CLK_FB 超前于 CLK_REF) .....	28
图 3-13 WINDOW 生成信号关系 .....	28
图 3-14 WINDOW 信号波形关系图 .....	29
图 3-15 基本延时单元 .....	29
图 3-16 延时链功能结构 .....	30
图 3-17 延时链自调整系统 .....	31
图 3-18 子延时链 SHIFT 调节示意图 .....	31
图 3-19 LDO 电路结构 .....	32
图 3-20 标准脉冲生成电路 .....	33

图 3-21 CLK0 生成电路 .....	33
图 3-22 CLK0 信号模拟 .....	33
图 3-23 CLK2X 生成电路 .....	34
图 3-24 分频电路结构图 .....	34
图 3-25 分频时钟信号模拟 (3.5 分频) .....	35
图 3-26 One-shot 逻辑与数字控制逻辑关系图 .....	36
图 3-27 One-shot 计算过程 .....	36
图 4-1 FPGA 整体仿真电路模型 .....	38
图 4-2 DLL 模型仿真电路结构 .....	39
图 4-3 分布式 RC 网络模型 .....	39
图 4-4 Window 信号频率扫描 (60MHz -> 500MHz 正常响应) .....	41
图 4-5 Up_down 信号频率扫描 (60MHz -> 500MHz 正常响应) .....	41
图 4-6 LDO 输出电压 VOUT .....	42
图 4-7 LDO 交流仿真图 .....	42
图 4-8 高频模式下 LDO 抖动影响 .....	43
图 4-9 低频模式下 LDO 抖动影响 .....	43
图 4-10 输入时钟为 10ns, DIV 配置成 1.5 的输出波形图 .....	44
图 4-11 输入时钟为 10ns, DIV 配置成 2 的输出波形图 .....	44
图 4-12 输入时钟为 12ns, DIV 配置成 2.5 的输出波形图 .....	45
图 4-13 输入时钟为 10ns, DIV 配置成 3 的输出波形图 .....	45
图 4-14 输入时钟为 10ns, DIV 配置成 4 的输出波形图 .....	45
图 4-15 输入时钟为 10ns, DIV 配置成 5 的输出波形图 .....	45
图 4-16 输入时钟为 10ns, DIV 配置成 8 的输出波形图 .....	46
图 4-17 输入时钟为 10ns, DIV 配置成 16 的输出波形图 .....	46
图 4-18 输入时钟为 10ns, HFDLL 模式下 DIV 配置成 1.5 的输出波形图 .....	46
图 4-19 输入时钟为 12ns, HFDLL 模式下 DIV 配置成 4 的输出波形图 .....	47
图 4-20 输入时钟占空比为 3:7, DIV 配置成 4 的输出波形图 .....	47
图 4-21 数模混合仿真流程图 .....	48
图 4-22 OSDLL 与传统 DLL 锁定时间比较 .....	49
图 4-23 传统 DLL 与 OSDLL 各频段锁定时间对比 .....	49
图 4-24 OSDLL 版图 .....	50
图 4-25 测试样片照片 .....	51
图 5-1 DLL 和 FPGA 时钟网络架构 .....	52
图 5-2 DLL 板级时钟信号调节 .....	53
图 5-3 DLL Ring Oscillator (环震) 模式 .....	53
图 5-4 RO 测试模式系统 .....	54

图 5-5 RO 功能仿真波形 .....	54
图 5-6 RO 后仿波形 .....	55
图 5-7 FPGA 内 SEU 效应 .....	55
图 5-8 TMR 示意图 .....	56
图 5-9 FPGA 的 TMR 加固技术.....	56
图 5-10 DLL 的 TMR 加固技术.....	57

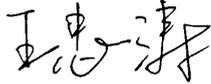
## 表格清单

表 4-1 高/低频模式下的可调时钟频率范围 .....	40
表 4-2 高/低频模式下的可调时钟锁定时间 .....	40
表 4-3 高/低频模式下的动态功耗 .....	40
表 4-4 仿真模型电路延时值 .....	40
表 4-5 FPGA 芯片可用逻辑资源 .....	50

## 独创性声明

本人声明所提交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标志和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签字：



签字日期：2010年4月16日

## 学位论文版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅或借阅。本人授权合肥工业大学可以将学位论文的全部或部分论文内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文者签名：



导师签名：



签字日期：2010年4月16日

签字日期：2010年4月16日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编：

## 致 谢

在本阶段论文完成、即将答辩、毕业之际，我谨向培养了我的学校和所有曾给予过我关怀、支持的人表示深深的谢意。

首先需要感谢的是我的导师杨明武老师。杨老师渊博的知识、深厚的素养、严谨的学术态度、感人的敬业精神以及勤勉的工作作风都使我受益匪浅，这些都给予我无限求知的动力，鞭策我不断前进，这将对我以后的学习、工作产生深远的影响。

感谢 SSMEC 公司设计四部的包朝伟，刘志刚，何文明工程师，指导我完成毕业设计，向我传授作为一位技术科研人员搞技术研究的经验，百忙之中抽出时间帮我解决一些实际问题，时刻关注我的设计进度并提出了自己宝贵意见。

感谢 SSMEC 公司设计四部的经理和同仁，为我提供了实习的机会，在生活上、工作上提供了非常周到的安排。而且在整个实习过程中，时刻关注我毕业设计的进度。他们治学严谨，工作一丝不苟，为人和蔼，深深地感染着我，整个毕业设计过程我学到的不光有知识，更有工作学习的态度以及团队合作的精神。

感谢实习单位的李健，刘宇铮，林钰凯和张弢等同事，在实习和完成毕设阶段所给与我的帮助和支持。

感谢合肥工业大学 07 级研究生 34 班的各位同学们，尤其是实验室的同学，于攀，刘勃，许银生，张瑞，在整个研究生阶段的学习过程中，给了我许多支持和帮助。

最后我要对我的父母表示深深的感谢。在他们的爱和支持下，我平静而充实的度过了象牙塔的生活，顺利的完成我的学业，有了资本和勇气开始书写自己的人生篇章。我很难想像父母究竟付出了怎样的艰辛和汗水才支撑我完成学业，我只想祝福父母生活幸福，健康快乐，我会用行动报答亲恩！

作者：王忠涛

2010 年 4 月

# 第一章 绪 论

## 1.1 论文的研究背景

FPGA(现场可编程逻辑阵列)芯片已广泛的应用在电路设计的各个领域。微电子技术的持续发展使得FPGA具有更高的系统集成度和工作频率,这也使得片内时钟管理的质量变得尤为重要。因为系统性能很大程度上决定于系统的时钟延迟和偏斜。由于FPGA具有丰富的可编程逻辑资源以及丰富的时钟网络,随之可能带来的时钟延迟问题将使得用户设计的性能大打折扣。传统的时钟树网络方法不再能够满足FPGA芯片的设计要求,这就需要新的时钟管理方法。

现代的时钟管理技术主要是基于PLL技术和DLL(Delay-Locked Loop, 延时锁相环)技术。PLL是常用的时钟管理电路,主要是基于模拟电路设计实现的,而DLL主要是基于数字电路设计实现的。由于数字DLL具有功耗小、灵敏度高、工艺可移植性好、不累计相位误差、设计仿真周期更短、抗干扰性更强等优点,所以项目中采用数字延时锁相环来实现FPGA片内时钟的管理。

延时锁相环(Delay Locked Loop)技术可以很好的提供FPGA片内时钟管理。延时锁相环具有功耗小、工艺可移植性好、不累计相位误差、设计仿真周期更短、抗干扰性更强等优点,虽然在频率综合能力上稍逊于PLL,但因为其主要基于数字电路设计完成非常适合作为FPGA内的时钟管理模块。

有关锁相技术最早的论述是1932年Bellescize提出的,目的是为了了解决同步检波如何得到本地振荡信号的问题。锁相技术基本上同电路设计的历史一同发展而来。近30年代以来,随着集成电路制造工艺技术的长足发展,锁相环电路逐渐成为了一种成本低廉、使用简便的时钟管理功能模块,为锁相技术在IC领域的广泛应用提供了机会和条件。目前,锁相技术已经被广泛的应用在IC芯片片内时钟调节管理,FSK解调、频率合成方面的应用以及FM解码等多种领域。

随着IC行业内的SOC, SoPC技术以及IP软硬核技术的发展。作为一个基本的库宏单元模块,在无线通讯和微处理器电路中,锁相环作为时钟管理电路得到了极其广泛的应用,此方面的研究也不断深入。一方面,在经典PLL架构中提出很多新的、性能优秀的单元模块电路,主要包括在压控振荡器、鉴频鉴相器的设计等方面;另一方面,锁相技术的电路结构也有了长足的发展,不再限于PLL的简单原始结构,延时锁相环DLL(Delay Locked Loop),混合模式延时锁相环MDLL(Mixed-mode Delay Lacked Loop)以及SMD(Synchronous Mirror Delay)等技术<sup>[1]</sup>也不断出现使得锁相技术迅猛的发展。

PLL(Phase Locked Loop)技术经过改进便可构建出延时锁相环(Delay Locked Loop)的架构原型。DLL继承了PLL电路的部分锁相技术,例如鉴相,时钟综合技术等,但去除了PLL电路的VCO振荡器部分,取而代之的是一条可调延时值的延时链<sup>[2]</sup>。进一步对PLL电路和DLL电路进行改进,使用数字电路单元代替其中的模拟器件就得到了全数字PLL电路<sup>[3]</sup>和全数字DLL电路<sup>[4]</sup>。

多数时钟管理电路都是基于PLL电路设计的。但是，随着延时锁相环DLL技术的逐渐发展和成熟，芯片内基于DLL电路的时钟管理系统已得到越来越多的应用。目前来说，低高频时钟转换电路(Frequency Multiplier-FM)设计难度较大，这是制约DLL作为时钟管理电路在IC设计领域内的广泛应用的一个重要因素。目前业界所使用的低高频时钟转换电路绝大多数没办法达到5倍以上的高倍数的倍频系数，一些倍频系数较高的电路结构又无法保证输出高频时钟的占空比。就目前的情况来说，多数基于DLL的时钟产生电路中，可控延时链大都采用模拟电路实现的。例如使用电荷泵电路、低通滤波器和压控电阻等模拟电路结构。模拟电路部分不仅给电路设计和芯片生产带来很大的困难，而且也直接限制了电路的可重用性。

如何通过数字电路设计方式依托数字电路流程完成一款延时锁相环的设计，使其具有较强的时钟综合能力满足芯片时钟管理要求，同时在较宽的输入时钟频段内能够完成锁相调整，并且不仅能够作为专用集成电路的嵌入模块，同时能够作为可编程逻辑器件，如FPGA芯片中的时钟管理功能块，将对DLL的设计提出新的挑战。

## 1.2 论文的研究目的和意义

目的：论文基于某IC设计公司的重点科研攻关项目SRAM型FPGA芯片设计，设计完成FPGA片内延时锁相环，连同FPGA时钟树结构配合搭建片内时钟管理系统，并在此基础上提出具有更快锁定速度的新延时锁相环架构。

(1) DLL功能目标：能够在较宽的频率范围内完成时钟同步调节，可以完成高频/低频不同输入时钟的调节，可以生成多种相移时钟，分频时钟，倍频时钟。DLL具有可配置性能够嵌入FPGA芯片内部，配合时钟树结构实现FPGA时钟管理功能，测试功能等。

(2) 基于原DLL架构，设计实现更快锁定速度的延时锁相环。结合延时计算技术，融合原DLL时钟调节工作机制设计新型DLL。

意义：在FPGA芯片设计领域，国外公司处于垄断地位，国内的现场可编程逻辑器件芯片设计处于空白，如何实现在这个领域的突破，填补国内可编程逻辑IC产业的空白将是极具研究及市场价值的。

在锁相环技术方面，国内已有一定程度的技术积累，但是可编程器件芯片数字延时锁相环的设计在国内研究尚处于起步阶段，尤其是FPGA片内的延时锁相环。因此本论文具有相当重要的价值和意义。

## 1.3 国内外研究进展

### 一、 国外的发展近况

现今，国外最新的FPGA已经发展至亿门量级电路的规模，芯片工作速度可

以达到GHz,其内部拥有海量的可编程资源,并且集成了大量的软核及硬核能够方便的提供用户使用。在同一芯片的实现方案中FPGA开始在多方面向ASIC提出了挑战,比如在设计可达到的规模、所用的设计时间、实现的功能以及成本控制上等。然而之所以FPGA能够发展到如此大的芯片规模和如此的强势市场占有率,无不与微电子行业不断发展的工艺技术以及先进高效的时钟管理技术联系在一起。FPGA 产业中的占统治地位的两大公司: Xilinx和Altera。这两个公司在其FPGA芯片中都集成了多个时钟管理模块用于处理用户设计高速运行时各种时钟的偏斜、延迟等问题<sup>[5][6]</sup>。

## 二、国内的发展近况

目前在国内,FPGA在市场中具有强大的竞争力以及广泛的应用前景,但是可编程器件产品和关键技术都被国外公司所垄断。国内除了少数院校如复旦大学微电子和公司对于FPGA的整体架构和内部的部分结构进行了研究性的探索外,基本上处于一片空白。目前刚开始起步,正在努力实现这一领域的长足发展和突破,在技术与水平上同国外的差距是十分巨大的。但与以往比较,国内现在已有比较先进的工艺制造技术和优秀的代工厂为设计实现FPGA提供了必要的工艺与技术支持。

## 三、可编程逻辑器件的发展趋势

目前先进的ASIC生产工艺已经被广泛的用于FPGA的生产,越来越多的处理器内核被集成到高端的FPGA芯片中,基于FPGA的开发和设计成为一项系统级的设计工程<sup>[7]</sup>。随着半导体工艺的不断提高,FPGA集成度将不断提高,同时制造成本将不断降低。FPGA的灵活性,可以与ASIC结合提升功能,所以它将成为实现系统集成的一种重要途径。

### 1. 具有大容量、低电压、低功耗的FPGA

大容量FPGA是市场发展的方向。FPGA 产业中的两大企业: Altera和Xilinx在大容量FPGA上展开激烈的竞争。2007年Xilinx推出的65nm工艺的Vortex IV系列芯片,容量为33792个Slices,Altera推出了65nm工艺的Stratix III系列芯片,容量为67200个逻辑单元。采用深亚微米(DSM)的工艺的逻辑器件芯片在性能提高的同时价格逐步降低。由于便携式应用产品的发展,对FPGA的低功耗的要求日益迫切。因此无论那个厂家、哪种类型的产品,都在向这个方向努力。

### 2. 高密度系统级FPGA

随着生产规模的扩大,产品应用成本的下降,FPGA 的应用已不是过去仅适合于系统接口部件的现场集成,而是将它灵活地应用于系统级(包括其核心功能芯片)设计。在此背景下,国际主要FPGA厂家在高密度系统级FPGA技术发展上主要强调了两个方面:FPGA的IP(Intellectual Property)硬核和IP软核。当前,具有IP内核的系统级FPGA的开发主要表现在以下两个方面:一是FPGA厂商将IP硬核嵌入到FPGA 器件中;另一方面是大力扩充优化的IP软核,即利用HDL

语言设计并经过综合验证的功能单元模块。用户可以利用预定义的、经过测试和验证的IP核，有效地完成复杂的SOC设计。

### 3. FPGA和ASIC相互融合

虽然ASIC芯片具有尺寸小、功能强、功耗低等特点，但其设计复杂，并且有产量要求。FPGA价格较低廉，能现场进行编程，但芯片面积较大、性能有限，而且功耗比ASIC大很多。正是因为如此，FPGA和ASIC正在互相融合，更具两者的特点取长补短。随着一些ASIC制造商提供具有可编程逻辑的标准单元，FPGA制造商重新对半定制的标准逻辑单元产生兴趣。

### 4. 动态可重构FPGA

动态可重构FPGA<sup>[8]</sup>是指在一定条件下芯片不仅具有系统重新配置电路功能的特性，而且具有系统动态重构电路逻辑的能力。对于数字时序逻辑系统设计，动态可重构FPGA的意义在于其时序逻辑的发生不是通过调用芯片内不同区域、不同逻辑资源来组合而成，而是通过对FPGA进行局部的或全局的芯片逻辑的动态重构来实现的。动态可重构FPGA在器件编程结构上具有专门的特征，类似于ASIC，其内部逻辑块和内部连线的改变，可以通过读取不同的配置点SRAM中的数据来直接实现这样的逻辑重构，时间往往仅需纳秒级，有助于实现FPGA系统逻辑功能的动态重构。

### 5. EDA工具的升级

EDA工具正朝着越来越人性化的设计，越来越高的优化水平，越来越快的仿真速度，越来越高的仿真精度以及完备的分析验证手段的方向前进<sup>[9]</sup>。

## 1.4 本论文主要的内容及结构

论文主要介绍FPGA芯片片内延时锁相环研究和设计。论文基于某IC设计公司的FPGA芯片设计项目中的DLL模块设计，根据DLL的模块化结构分别完成半定制和全定制设计，流片完成一款100万门SRAM型FPGA芯片，测试结果显示FPGA的DLL时钟管理功能正确，符合设计要求。

具体各章的主要内容如下：

第一章 绪论 主要阐述锁相技术的发展背景及FPGA芯片的国内外研发现状和发展趋势。介绍了可编程器件中的时钟管理技术采用DLL(延时锁相环)技术的优点以及完成FPGA片内延时锁相环设计的重要意义。

第二章 FPGA片内DLL工作原理 介绍了DLL在FPGA内部的工作机制和DLL结构设计原理，在原有DLL设计的基础上引入one-shot延时计算及调整概念，提出新的延时锁相环OSDLL架构，并阐述了OSDLL工作原理。

第三章 FPGA片内DLL架构设计 阐述了DLL整体模块架构，包括鉴相器，数字控制逻辑，可调延时链，时钟生成模块等。并分别详细阐述了各个子模块的电路结构设计。在原DLL架构设计的基础上介绍了OSDLL架构设计。

第四章 DLL功能与性能仿真及流片结果 介绍了DLL参数仿真（SPICE）及数字功能仿真，并阐述了DLL半定制流程模块和全定制流程模块的混合仿真。最后将介绍SRAM FPGA的测试样片及结果。

第五章 DLL模块多种环境下的应用 介绍了DLL能够完成的主要功能和相应的应用领域，同时论述了FPGA，特别是DLL在太空辐照情况下的抗SEU（单粒子翻转）设计。

第六章 结论 总结论文体系和设计结果，指出未来的研究方向。

## 第二章 FPGA 片内 DLL 工作原理

### 2.1 锁相环在 FPGA 时钟网路中的作用

同步数字系统，包括板级系统和芯片级系统，依靠一个或多个时钟信号来同步系统内的各单元。一般来说，这样的—个或多个时钟信号通过一个或多个时钟线传播到系统各个单元。然而由于各种原因，例如时钟 buffer 延时，重负载时钟线的高电容值，传播延时等原因，使得时钟上升沿在系统各部分也许不是同步的。在系统某一部分的时钟上升沿同另一部分的时钟上升沿之间的差距称作“时钟偏斜” [10]。

时钟偏斜会导致数字系统发生故障。例如，常见的一种情况，数字系统中一个触发器的输出驱动第二级触发器的输入。如图 2-1，FF1 和 FF2 两个触发器的时钟端都接入同一个同步时钟，则第一级触发器的数据能够成功的传递到第二级触发器。然而，如果第二级触发器的时钟有效沿因为时钟偏斜延迟了，那么可能在第一级触发器改变状态前，第二级触发器会采不到数据。

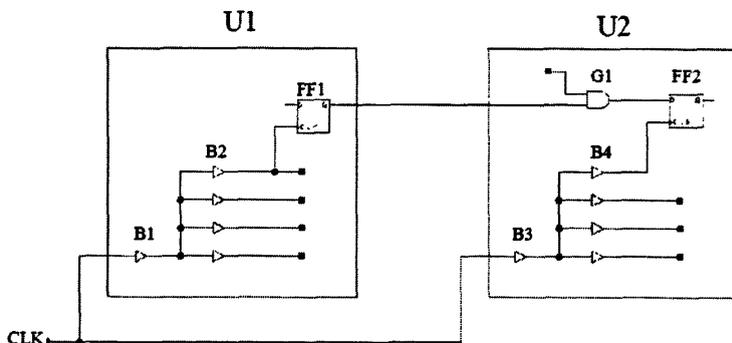


图 2-1 DLL 时钟管理举例

随着大规模 FPGA 芯片的研发，片上时钟在很大程度上影响着芯片和系统的整体性能。传统时钟树的时钟线分布造成的时钟偏斜和延迟使得大规模器件 FPGA 设计出现难题。采用片内增加可配置的延时锁相环在很大程度上解决了这个难题。延时锁相环消除了时钟延时，减小了时钟偏斜。使输入时钟和时序电路的到达时钟同步。同时延时锁相环还能够生成原时钟的多版本相移，分频，倍频时钟用以 FPGA 片内及板级设计需要。

### 2.2 DLL 与 PLL 比较

时钟同步的方法通常是通过 PLL 和 DLL 来实现的。PLL 通常是由模拟电路实现的，需要很长的时间仿真，而且在某个工艺下生产的芯片，在改变工艺后很可能会功能失效。其外，模拟 PLL 对于辐射是非常敏感的。所以 PLL 是非常难于设计，可移植性差的电路设计。PLL 和 DLL 基本架构 [11] 比较如图 2-2:

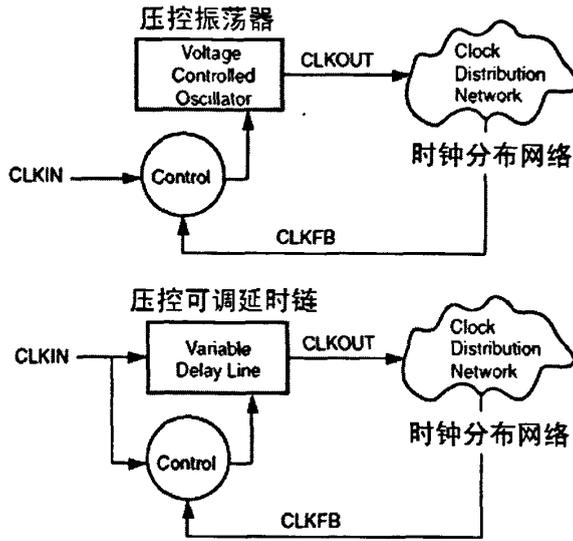


图 2-2 DLL 与 PLL 简化结构对比图

PLL 的结构主要由四个部分组成，即鉴相器（Phase Detector, PD）、环路滤波器、压控振荡器(VCO)和频率除法器（Frequency Divider）<sup>[12][13]</sup>。

鉴相器：PD 用来比较 PLL 的输出相位和输入相位，其平均输出电压为一个与两个输入时钟的相位差成线性比例的直流电压。鉴相器基本上分为数字鉴相器和模拟鉴相器两类。

低通滤波器：在实际的电路中，鉴相器的输出不仅包括所希望的直流分量还包括不希望的高频分量，而振荡器的控制电压在稳态时必须保持恒定，所以必须采用一个低通滤波器来滤除其中的高频分量，即在鉴相器和 VCO 振荡器之间插入一个低通滤波器来抑制高频成分，把直流分量送到振荡器。

压控振荡器：它是一个电压-频率变换电路。在 PLL 中作为振荡器，其振荡频率随输入控制电压线性地变化，这是锁相环设计的核心。

延时锁相环 DLL 的结构与 PLL 相似，只是用压控可调延时链 VCDC（Voltage Control Delay Chain）代替了压控振荡器，如图 2-2 下半图。

压控可调延时链是由一系列电压控制的延时单元串联而成的链条，输出信号是输入信号的延迟版本，把压控可调延时链的输入和经过延迟后的输出送入鉴相器中进行比较，通过锁相环使两者之相差锁定在一个周期（同相比较）每个延时单元的延迟时间就为  $T/n$ ，其中  $n$  为延迟的级数。

PLL 中的 VCO 的输出频率和控制电压成正比，频率是相位的积分。因此传输函数有一个极点  $sKVCO$ 。而对于 VCDC，其输出相位与控制电压成正比，传输函数是一个常数。对于一阶环路滤波器，DLL 系统传输函数是一阶方程，相对于 PLL 二阶系统来说（通常来说还要外加一个旁路滤波器构成一个三阶系统），增益、带宽、系统稳定性考虑将更加容易。因此 DLL 常用来生成稳定的

延迟或者多相位的时钟信号。

## 2.3 DLL 工作原理

### 2.3.1 DLL 原理概述

在数字系统中，DLL 用以减小时钟偏斜。DLL 通常使用延时单元来同步参考时钟和反馈时钟。图 2-3 所示为 DLL 简化结构框图<sup>[14]</sup>，DLL 包括以下几部分：延时链，鉴相器。DLL 接收参考时钟 REF\_CLK，驱动一个输出时钟 O\_CLK。

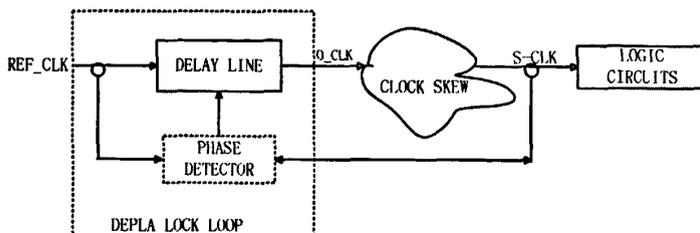


图 2-3 DLL 简化结构框图

延时链在输出时钟信号 O\_CLK 之前，首先将参考时钟 REF\_CLK 延迟一定的时延 D。这样每一个输出时钟的边沿都比参考时钟推后 D。鉴相器控制延时链。延时链能够产生一个最小的传播延时 D\_MIN 和一个最大的传播延时 D\_MAX。

在输出时钟信号 O\_CLK 达到逻辑电路之前，输出时钟 O\_CLK 会发生时钟偏斜。偏斜的发生可能是因为各种时钟 buffer 缓冲或者是时钟线传播延时产生的（时钟线重负载）。为区别 O\_CLK 与发生偏斜后的 O\_CLK，我们用 S\_CLK 来表示发生偏斜后的时钟信号。S\_CLK 驱动逻辑电路的时钟端。同时 S\_CLK 从反馈路径反馈回 DLL。一般来说，反馈路径是 S\_CLK 的专属路径（FPGA 时钟线设计要求）。所以任何在反馈路径上产生的延时是很小的，并且只会产生负偏斜。

图 2-4 为参考时钟 REF\_CLK 和输出时钟 O\_CLK 以及偏斜时钟 S\_CLK。这三个时钟频率相同，周期为 P，频率为 F，并且都是上升沿有效。O\_CLK 延迟于 REF\_CLK 传播延迟 D，S\_CLK 延迟于 O\_CLK 传播延迟 SKEW(时钟偏斜产生的)。因此，偏斜时钟 S\_CLK 的时钟沿落后参考时钟 REF\_CLK 延迟  $DSKEW=D + SKEW$ 。

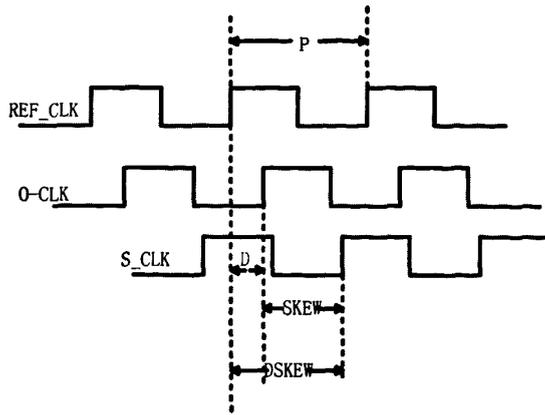


图 2-4 DLL 时钟调整前波形

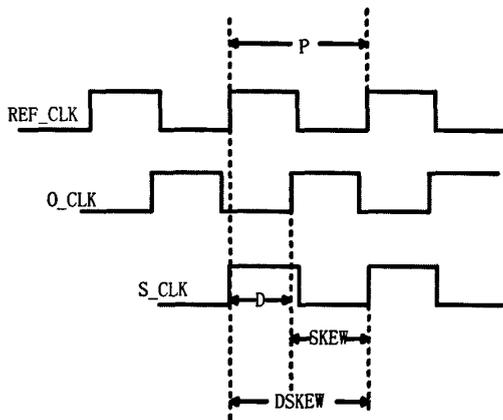


图 2-5 DLL 时钟调整后波形

DLL 通过控制延时链来调节传播延时  $D$ 。尽管延时链不能产生负的延时，但幸运的是，时钟信号是周期信号，所以 DLL 可以通过进一步的延迟输出时钟信号  $O\_CLK$  使得  $S\_CLK$  和  $REF\_CLK$  的时钟沿同步。如图 2-5 所示，通过调整传播延时  $D$ ，使得  $DSKEW = P$ ， $D = P - SKEW$ 。虽然  $DSKEW = mult(P)$ （多周期），都可以达到同步，但是大多数 DLL 受延时链产生的延时量的限制不能产生那么大的延时值。

鉴相器控制延时链生成传播延时  $D$ 。DLL 的具体控制机制可能会不同。例如，一种版本，上电复位后，延时链起初的延时  $D = D_{MIN}$ ，然后鉴相器增加延时值  $D$ ，直到参考时钟信号  $REF\_CLK$  同  $S\_CLK$  同步为止。在其他系统中  $D$  的初值取中间值，然后鉴相器增加或减少延迟  $D$  的值，使得两时钟达到同步。

超前 (lead)，滞后 (lag) 解释<sup>[15]</sup>：lag:  $REF\_CLK$  上升沿至下一  $S\_CLK$  上升沿小于  $S\_CLK$  上升沿至下一  $REF\_CLK$  上升沿。相反称为 lead。

将  $S\_CLK$  同  $REF\_CLK$  同步后，DLL 监测参考时钟  $REF\_CLK$  和  $S\_CLK$ ，通过调整传播延时  $D$  来保持同步。例如，如果传播延时  $SKEW$  增加，也许是由

工作温度升高引起的，DLL 必须减小传播延时  $D$  来补偿 SKEW。相反，如果传播延时 SKEW 减小，也许是温度降低引起的，DLL 必须增加传播延时  $D$ 。DLL 将 REF\_CLK 和 S\_CLK 调整至同步所需的时间段称为锁定获取（lock acquisition）。DLL 维持同步的时间段称为锁定保持（lock maintenance）。在锁定获取阶段结束时的传播延时值  $D$ ，也就是当同步调整刚刚完成时的  $D$  值，称为  $ID$ 。

从上边的解释，我们知道，延时链只能提供大于  $D_{MIN}$ ，小于  $D_{MAX}$  的传播延时。在锁定保持阶段，如果为保证同步所需的传播延时  $D$  小于  $D_{MIN}$  或者大于  $D_{MAX}$ ，则 DLL 将失去同步调整的能力。

例如，如果锁定获取阶段发生在系统处于高温的情况下，那么 DLL 很可能以一个非常小的初值传播延时值  $ID$  达到同步，因为 SKEW 相对于周期  $P$  可能很大。当系统温度进一步升高，SKEW 可能增大到  $SKEW + D_{MIN}$  仍然大于周期  $P$ 。在这种情况下，DLL 必须重新进行锁定获取，这样会在输出时钟信号 O\_CLK 中引入毛刺和噪声，也使得 S\_CLK 产生毛刺和噪声。对于关键性系统来说，这样的毛刺和噪声是不可接受的。另一方面，对于具有多时钟域的系统来说，低频时钟操作当时钟周期  $P$  很大时很可能出现问题。大时钟周期会导致传播延时  $D$  在很大的时间间隔中变化。这就使得 DLL 需要在很宽的时钟频谱和环境条件下具有保持同步的能力。

### 2.3.2 DLL 模块化结构及工作原理

DLL 包括有延时链，时钟相移器，控制器，和时钟生成模块，鉴相器，如图 2-6。DLL 接收参考时钟 REF\_CLK，生成输出时钟 O\_CLK。输出时钟 O\_CLK 经过偏斜后得到偏斜时钟 S\_CLK。偏斜时钟 S\_CLK 作为反馈时钟经过反馈路径反馈回 DLL。

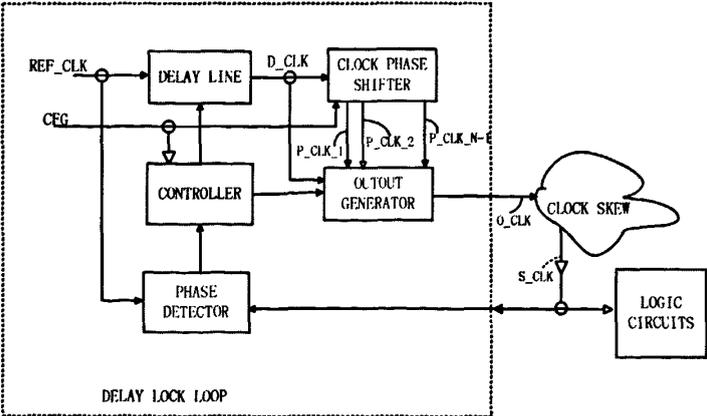


图 2-6 DLL 模块结构图

在 DLL 内部，延时链延时参考时钟 REF\_CLK，生成延时后时钟信号 D\_CLK，D\_CLK 是 REF\_CLK 经过延时链延时 D 后生成的。D\_CLK 送入时钟相移器和时钟生成器。

时钟相移器生成一个或多个相移后的时钟信号 P\_CLK<sub>1</sub>...P\_CLK<sub>N-1</sub>，N 是正整数。P\_CLK<sub>1</sub> 是相移 360/N 度的相移时钟，P\_CLK<sub>2</sub> 是相移 2\*(360/N) 度的相移时钟，P\_CLK<sub>N-1</sub> 是相移(N-1)\*360/N 度的相移时钟。P\_CLK<sub>Z</sub> 是相移 Z\*(360/N)度的相移时钟，Z 是 1 到 N-1 的整数。延时时钟信号 D\_CLK 可以认为是相移 0 度的相移时钟 P\_CLK<sub>0</sub>，因为相对 D\_CLK 自身来说 D\_CLK 具有 0 相移。进一步来说，DLL 可以生成具有同样频率的相移时钟 P\_CLK<sub>N</sub>。

在本设计版本中，N=4，相移时钟信号 P\_CLK<sub>1</sub> 是 D\_CLK 相移 90 度的相移时钟。之后同样，相移时钟信号 P\_CLK<sub>2</sub> 是 D\_CLK 相移 180 的相移时钟，相移时钟信号 P\_CLK<sub>3</sub> 是 D\_CLK 相移 270 移时钟。

相移是时钟信号在频域中的概念。在时域中，相移等价于时钟延时。具体来说，如果第一个时钟信号是第二个时钟信号相移 X 度得到的，也就是说第一个时钟比第二的时钟延迟了  $X*(P/360)$ ，P 是两时钟的时钟周期。所以，如果 P\_CLK<sub>1</sub> 是 D\_CLK 相移 90 度的相移时钟，P\_CLK<sub>1</sub> 也就是 D\_CLK 延时了 1/4 周期的延时时钟信号。P\_CLK<sub>Z</sub> 是相移 Z\*(360/N)度的相移时钟，也就是说 P\_CLK<sub>Z</sub> 具有相移延时  $Z*(P/N)$ ，Z 是 1 到(N-1)的整数。

当 N=4 时，P\_CLK<sub>1</sub> 为相移 90 度的相移时钟（延时 1/4 周期），P\_CLK<sub>2</sub> 为相移 180 度的相移时钟（延时 2/4 周期），P\_CLK<sub>3</sub> 为相移 270 度的相移时钟（延时 3/4 周期）。

时钟相移器将相移后的时钟输入至时钟生成模块。CFG 是配置信号，可控制控制器和时钟相移器。时钟生成模块，选择 D\_CLK(P\_CLK<sub>0</sub>)，或其中之一的相移时钟信号作为输出时钟信号 O\_CLK。例如时钟生成模块可以选择 P\_CLK<sub>N</sub> 作为时钟输出信号。

控制器从鉴相器接收参考时钟和偏斜时钟的相位信息。具体来说，鉴相器指示控制器延时链的传播延时 D 应该增加还是减小，以使得参考时钟 REF\_CLK 和 S\_CLK 同步。DLL 包含一个 jitter filter（抖动滤波），jitter filter 是一个加/减计数器，如果 D 应该减少时，则计数器减一；如果 D 应该增加时，则计数器加一。但是只有当加/减计数器达到 0 或某个设定的数值后，D 才会被调整。

锁定窗<sup>[16]</sup>W 的概念，如图 2-7，传播延时 D 的调整，一定要在 D\_MIN 和 D\_MAX 区间内。当前设计 D\_MIN 的典型值是 3.2ns，D\_MAX 的典型值是 46.8ns。在锁定获取阶段，控制器要确保 D 的初始传播延时值 ID 在锁定窗内。具体来说，当初步建立同步时，初始传播掩饰 ID 一定要在最小锁定窗 W\_MIN 和最大锁定窗 W\_MAX 之间。锁定窗 W 的限制是为了当 DLL 完成锁定获取后使得 DLL 能够保持同步。

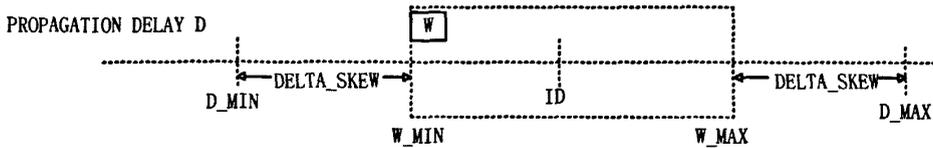


图 2-7 锁定窗图示

例如，系统含有一个 DLL 在一定的工作条件下工作。工作条件包括最极端的条件下，传播延时 SKEW 的最大值 SKEW\_MAX；同样具有一个传播延时的最小值 SKEW\_MIN。所以传播延时 SKEW 的最大变化量 DELTA\_SKEW=SKEW\_MAX - SKEW\_MIN。在锁定保持阶段，为达到最大保护的目的是锁定窗的最小值  $W\_MIN = D\_MIN + DELTA\_SKEW$ ， $W\_MAX = D\_MAX - DELTA\_SKEW$ 。在这一版本中锁定窗的最小值 W\_MIN 是最大传播延时 D\_MAX 的 16.5%，而 W\_MAX 是最大传播延时 D\_MAX 的 67.8%。

从图 2-3，可以看到对于传统 DLL，调整参考时钟 REF\_CLK 和偏移时钟 S\_CLK 之间同步，是通过使得传播延时

$$D+SKEW=MULT(P) \quad (1)$$

实现的，MULT(P) 是多周期，通常来说多周期 MULT(P) 最小值大于 SKEW。

本 DLL 控制器同样可以利用相移时钟信号的延时。所以如果 D 加上相移时钟产生的相移延时 P\_D 再加上传播延时 SKEW 等于多周期 P，则 DLL 能够达到同步，公式：

$$D+P\_D\_Z+SKEW=MULT(P) \quad (2)$$

P\_D\_Z 是相移时钟 P\_CLK\_Z 产生的相移延时。通常来说，多周期 P 的最小值要大于 SKEW 加上 P\_D\_Z。基于之前的解释，移时钟 P\_CLK\_Z 的相移延时 P\_CLK\_Z 等于  $Z*(P/N)$ ，Z 是 0 到 N-1 的正整数。如果 Z=0，控制器指示时钟生成模块使用 D\_CLK (P\_CLK\_0) 作为输出时钟 O\_CLK。所以相移延时等于 0。

为清楚起见，如果时钟生成电路使用 D\_CLK (P\_CLK\_0) 作为输出时钟 O\_CLK，那么初始延时 ID 称为 ID\_0。同样如果时钟生成电路使用 P\_CLK\_Z 作为输出时钟 O\_CLK，那么初始延时 ID 称为 ID\_Z，Z 是 0 到 N-1 的正整数。那么当完成锁定获取后，方程 (2) 可以重新写为：

$$ID\_Z + P\_D\_Z + SKEW = MULT(P) \quad (3)$$

重新写 (3)

$$ID\_Z = MULT(P) - P\_D\_Z - SKEW \quad (4)$$

用  $Z*(P/N)$  代替 P\_D\_Z 则得到：

$$ID\_Z = MULT(P) - SKEW - Z*(P/N) \quad (5)$$

通常来说，应使得 ID\_Z 是正值的多周期 P 的最小值。在仿真中，当 ID\_Z

小于最小传播延时  $D_{MIN}$  或者大于最大传播延时  $D_{MAX}$ ，则 DLL 不能用  $P\_CLK\_Z$  同步同步  $S\_CLK$  和  $REF\_CLK$ 。

因为控制器能够选择任意一个  $P\_CLK\_Z$  作为输出时钟信号  $O\_CLK$ ，所以理论上来说，控制器可以选择  $N$  个初始延时值  $ID$ 。可能的初始延时值的范围是最小值  $(MULT(P) - SKEW)$ ，至最大值  $(MULT(P) - SKEW + (N-1)/(N*P))$ 。每个初始延时值之间相差  $P/N$  的延时值。例如，如果  $N=4$ ， $P=40ns$ ， $SKEW=25$ ，那么初始延时值  $ID_0, ID_1, ID_2, ID_3$  分别等于  $15ns$ ， $5ns$ ， $35ns$ ， $25ns$ （基于公式 5 计算）。这样控制器可以找到一个或多个初始延时值落在锁定窗  $W$  内，控制器可以选择锁定窗  $W$  内的任意一个初始延时值。

控制器可以实现以上叙述的计算功能，决定使用哪一个相移时钟  $P\_CLK\_Z$ 。

### 2.3.3 子模块结构及工作原理

#### 2.3.3.1 时钟移相器及其工作原理

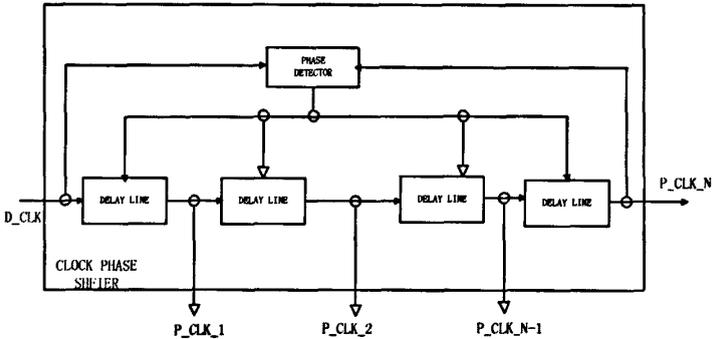


图 2-8 时钟移相器基本结构

图 2-8 是一种时钟相移器的模式。时钟相移器是由鉴相器，和 1 到  $N$  个延时链组成的。延时链 1 到  $N$  串连在一起。延时链的输入端接收时钟信号  $D\_CLK$ ，延时链的输出端同输入端一起接入鉴相器。鉴相器同时控制所有的延时链，每条延时链提供相同的的延时值。结果使得，输入时钟信号  $D\_CLK$  和时钟信号  $P\_CLK\_N$  在相位上同步。进一步来说，鉴相器使得 1 到  $N$  的延时链产生的总延时等于输入时钟一个周期值  $P$ 。这样每条延时链产生的传播延时为  $P/N$ ，时钟相移器的输出端输出与输入时钟信号相差  $P/N$ ， $2*P/N... (N-1)*P/N$  延时的时钟信号。通常来说，延时链的输出端输出相较于输入时钟延时  $Z*P/N$  的时钟信号， $Z$  是 1 到  $N-1$  的正整数。如果输入时钟信号是延迟时钟  $D\_CLK$ ，那么延时链的输出端为相移时钟  $P\_CLK_1$  至  $P\_CLK_{N-1}$ 。同时也会生成具有相同相位的时钟  $P\_CLK_N$ 。

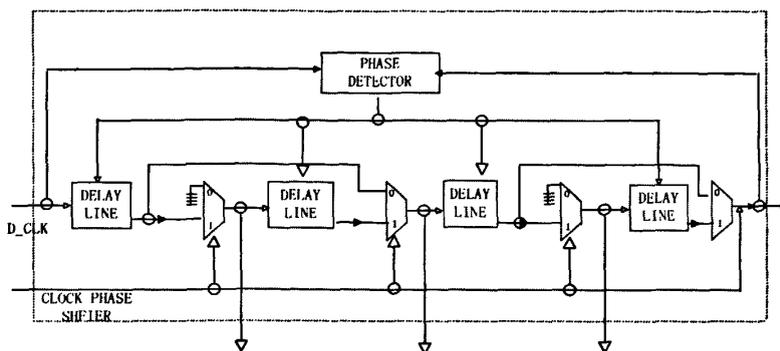


图 2-9 时钟移相器的可配置结构

图 2-9 为可配置的时钟移相器。具体来说，时钟移相器可够配置成两种模式。模式一，生成输入时钟的三个相移时钟信号 90 度，180 度，270 度。模式二，只生成一个相移时钟 180 度。时钟相移器包含一个鉴相器，四条延时链，和开关选择器。

如果配置点等于 1，则时钟相移器工作在模式一，每个延时链提供  $P/4$  延时。可输出 D\_CLK, P\_CLK\_1, P\_CLK\_2 和 P\_CLK\_3。

如果配置点等于 0，则时钟相移器工作在模式二，只有延时链 1, 3 串连。延时链 2, 4 的输入端接低电平，有助于减小功耗和开关噪声。在第二模式下，延时链 1, 2 分别提供  $P/2$  的延时。模式二只产生一个输出时钟，180 度。

### 2.3.3.2 时钟生成模块及其工作原理

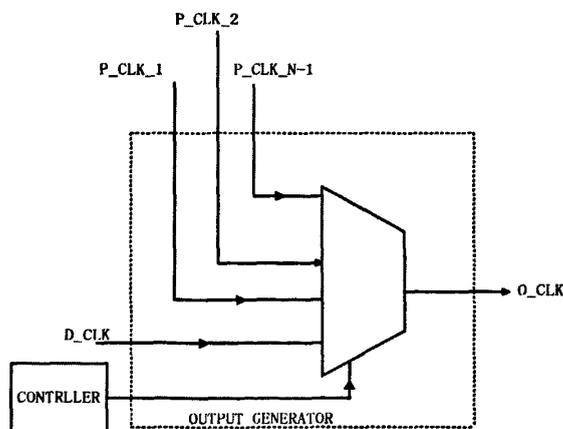


图 2-10 时钟生成模块原理示意图

图 2-10 是时钟生成模块<sup>[17][18]</sup>可以选择 D\_CLK, P\_CLK\_1, P\_CLK\_2 和 P\_CLK\_3 之一输出，配合控制器进行工作。同时含有分频倍频功能。

### 2.3.3.3 控制器及其工作原理

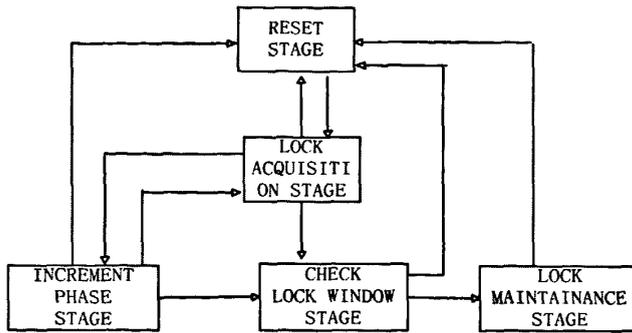


图 2-11 状态机控制器原理示意图

图 2-11 是控制器状态图，逻辑控制部分的状态机转移示意图。上电或复位后，控制器进入 reset 状态。在 reset 状态，控制器将相位计数器设置为 0，使得时钟生成模块选择 D\_CLK(P\_CLK\_0)作为输出时钟。控制器调整延时链的传播延时 D 至起点延时值。D 的起点延时值可以是 D\_MIN, D\_MAX 或两者的中间值，同样可以通过计算方式给出，在 2.3.2 节的 OSDLL 设计原理中将详细说明。控制器之后进入锁定获取阶段。

在锁定获取阶段(LOCK ACQUISITION)，控制器同步参考时钟 REF\_CLK 和偏斜时钟 S\_CLK。具体来说，控制器根据鉴相器的指示信号，调整延时链的传播延时。鉴相器决定传播延时应该加/减，以使得两时钟同步。在一种版本中，控制器等到时钟移相器生成相移时钟 P\_CLK\_1 到 P\_CLK\_N-1 之后，才开始调整传播延时 D。如果控制器不能同步参考时钟 REF\_CLK 和偏斜时钟 S\_CLK，控制器将进入相位增加状态(INCREMENT PHASE)。控制器将两时钟调整至同步后，进入锁定窗检测状态(CHECK LOCK WINDOW，具有一个初始传播延时 ID)。

在锁定维持阶段(LOCK MAINTAINANCE)，控制器调整传播延时 D 以保持 S\_CLK 和同 REF\_CLK 同步。目前版本的 DLL 能够在系统环境条件下维持锁定，除非 reset 信号将控制器跳转至 reset 状态。

在相位增加状态，控制器增加相位，相位计数器执行加操作，使得时钟生成模块选择不同的相移时钟信号。进一步来说，控制器复位延时链，使得传播延时 D 至 reset 状态的起点延时值。控制器然后进入锁定获取状态，继续以上操作。

## 2.3.4 OSDLL 工作原理

### 2.3.4.1 one-shot 原理

One-shot<sup>[19]</sup>调整方式是基于以下的电路结构实现的，如图 2-12。结构包括延时链，译码电路和控制电路。

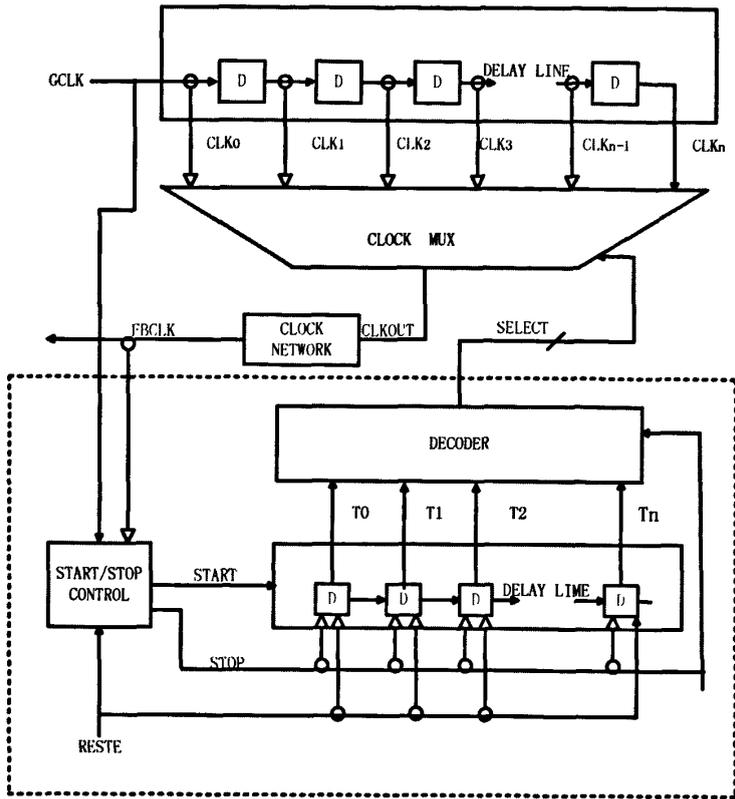


图 2- 12 one-shot 原理结构示意图

One-shot 基本原理，如图 2-13，为同传统 DLL 原理介绍做区别，这里取 GCLK 为输入时钟，FBCLK 为反馈时钟。

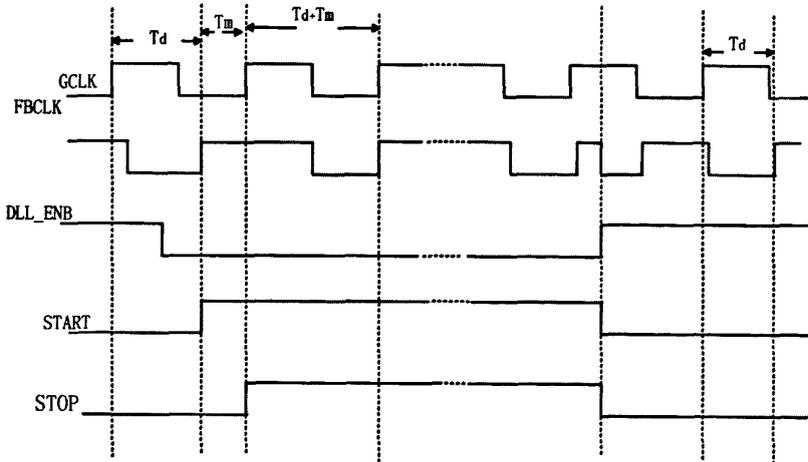


图 2- 13 one-shot 原理波形示意图

GCLK 上升沿同 FBCLK 上升沿有  $t_d$  大小的延时，则  $T_m = P - T_d$  即消除  $T_m$  便可以使得两时钟同步。START 信号同 CLKFB 上升沿（下降沿）同步，STOP

信号同 CLKIN 上升沿（下降沿）同步，如图 2-13。START（恒“1”）信号送入延时链的输入端，每个延时单元的输出端作为译码电路的输入信号，STOP 为译码器的采样信号，即 STOP 信号采样到类似“111100000...”的采样值。其中“1”的个数代表了  $T_m$  延时单元的数目。通过译码电路计算出两者之间的延时单元数目，即  $T_m$  粗略等于多少个延时单元。完成计算后，选择经过相应延时的 GCLK 延时版本时钟作为调整后的反馈时钟信号 GCLKZ，其中  $Z(1\sim N)$ ，输入到时钟网络中。图 2-12 中，实际上 one-shot 工作方式只用了一条延时链，但为方便说明图中画出两条，是不同工作阶段延时链工作示意图。

DLL 接收输入时钟信号和反馈时钟信号，通过加入一定的 delay 来同步两时钟。只需要一步简单操作，即可实现同步。同时只需要一条延时链，用来实现 DLL 电路。所以这一设计的 DLL 有着锁定速度快，硬件开销小的优点。另外，这一 DLL 也是十分精确的。

此版本的 DLL 是由输入时钟端口，反馈时钟端口，延时链，控制电路组成的。DLL 工作过程分两个步骤。步骤一计数（counting），反馈时钟输入延时链，延时链的一串延时单元的输出端，也就是反馈时钟经过一级级延时单元延迟后的中间时钟信号，各中间时钟信号作为译码器的输入。步骤二操作（operating），采样信号采样步骤一中的一组译码输入值，这组值被锁存。译码电路计算出锁存值代表的延时单元数目。

译码器用以计算反馈时钟上升沿和相邻下一输入时钟上升沿之间的 delay，通过延时单元的数目换算出 delay 值。然后将经过相应延时单元数目的反馈时钟作为调整后时钟送入时钟网络。

One-shot 调整方式最大的好处是可以单节拍实现相位同步调整，一旦完成译码之后，即选定相应的输出时钟。然而这一方式的调整精度严重的依赖延时单元的最小延时，如果单位精度比较大，则同步调整的效果也比较差，而且由于没有防抖动电路支持，其稳定性也不高。同时当输入时钟和反馈时钟的 skew 比较大时，需要译码电路检测一条很长的延时链的每一个延时单元的输出信号（比如 256 个输出信号），这势必造成需要超长位数的译码电路，这对于电路性能和延时要求又提出了新的问题。

#### 2.3.4.2 OSDLL 原理

2.3.1 介绍的是传统 DLL 架构下的设计，其具有设计周期相对较短，工艺可移植，抗干扰能力强等特点。但同时由于其控制逻辑的工作特点，从复位状态开始，延时链初始值 D 复位至 0，即可逆计数器从 0 开始计数。DLL 按照工作节拍信号，一拍一拍的进行调整（假设六周期一节拍），当输入时钟频率较低或者时钟相差较大时，其锁定时间将大大增加。取任何固定值作为延时链的复位值，同样存在某一频率段锁定时间较长的问题。D 初始值的静态设定无法满

足实际应用中多种 SKEW 造成的 ID 值迅速落入锁定窗的 DLL 同步目标。

One-shot 技术为这一问题的解决提供了很大的帮助。可以通过在同步获取阶段之前，首先粗略的计算出一个 D 初值，将其作为延时链初值，如图 2-14。SHIFT 为时钟移相器工作阶段，SYN 为同步获取阶段。

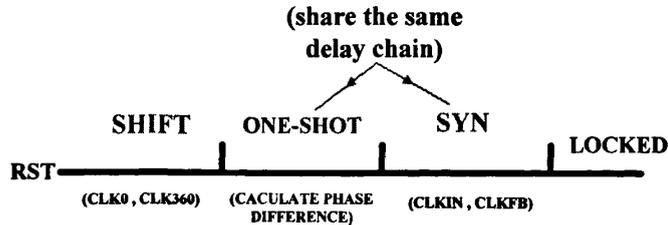


图 2-14 OSDLL 时钟调整过程

针对提出的问题，采用 one-shot 延时计算机制，即完成时钟相移器调整后，首先利用主延时链来计算反馈时钟的上升沿和输入时钟的上升沿之间的相位差值（假设有效沿是上升沿），这个延时值以延时单元的数目来表征。计算技术，以 one-shot 计算技术为基础，由一个专用译码电路将计算后的延时值作为延时链 D 初值，即经过译码后使得主延时链具有一个合理的延时初值。然后 DLL 进入锁定获取阶段，按照上述的过程进行同步调整。由于大部分的相差在 one-shot 计算结果付给可逆计数器时已经消失了，DLL 只需经过很短的调整周期即可达到同步。这种结构的 DLL，称之为 OSDLL。

## 2.4 本章总结

本章介绍了 DLL 在 FPGA 中的主要用途，并同 PLL 做了比较，DLL 的多种优点使其在可编程器件中作为时钟调整模块更加有利。后文继续介绍了 DLL 的工作原理，包括模块化的结构组成，实现锁定的方式和子模块的结构原理等。提出了一种新的延时锁相环结构，OSDLL。介绍了这种结构基于 one-shot 原理，将这种技术结合原有 DLL 原理技术可以更加快速的实现 FPGA 片内时钟调整锁定。

# 第三章 FPGA 片内 DLL 架构设计

## 3.1 DLL 架构概述

图 3-1 为 FPGA 片内 DLL 结构框图。图 3-1 中 FPGA 片内用户设计的时序逻辑部分在布局布线后位于芯片中部，相应的时钟走线较长。为缓解时钟缓冲，重负载时钟线的大电容，线路的传播延时等因素造成的时钟偏斜，可以选择使用 DLL 模块进行时钟优化管理。

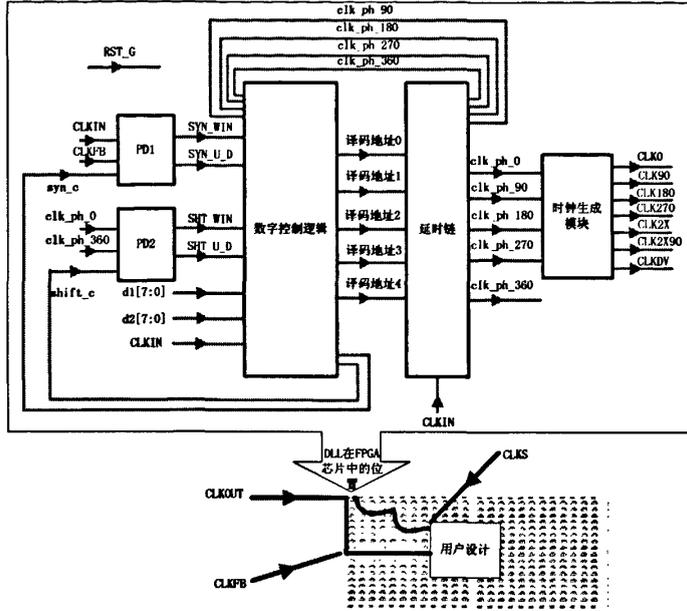


图 3-1 FPGA 芯片中 DLL 结构框图

图 3-1 中 DLL 主要由鉴相器(PD)，可调延时链，数字控制逻辑以及时钟生成模块组成。CLKOUT 为 DLL 输出时钟，即时钟生成模块的输出时钟；CLKS 为经过时钟线后到达时序电路的偏斜时钟；CLKFB 即为 CLKs，反馈时钟 CLKFB 反馈回 DLL。DLL 的功能为通过在时域中调节 CLKOUT 的相位使得 CLKFB 同 CLKIN 同步，即消除时钟偏斜。

## 3.2 数字控制逻辑设计

DLL 的工作过程依赖于控制逻辑的设计。DLL 的控制逻辑主要包括 SHIFT 控制逻辑和 SYN 控制逻辑两部分，如图 3-2 所示。DLL 的工作过程首先进行 SHIFT 阶段，之后进行 SYN 阶段。

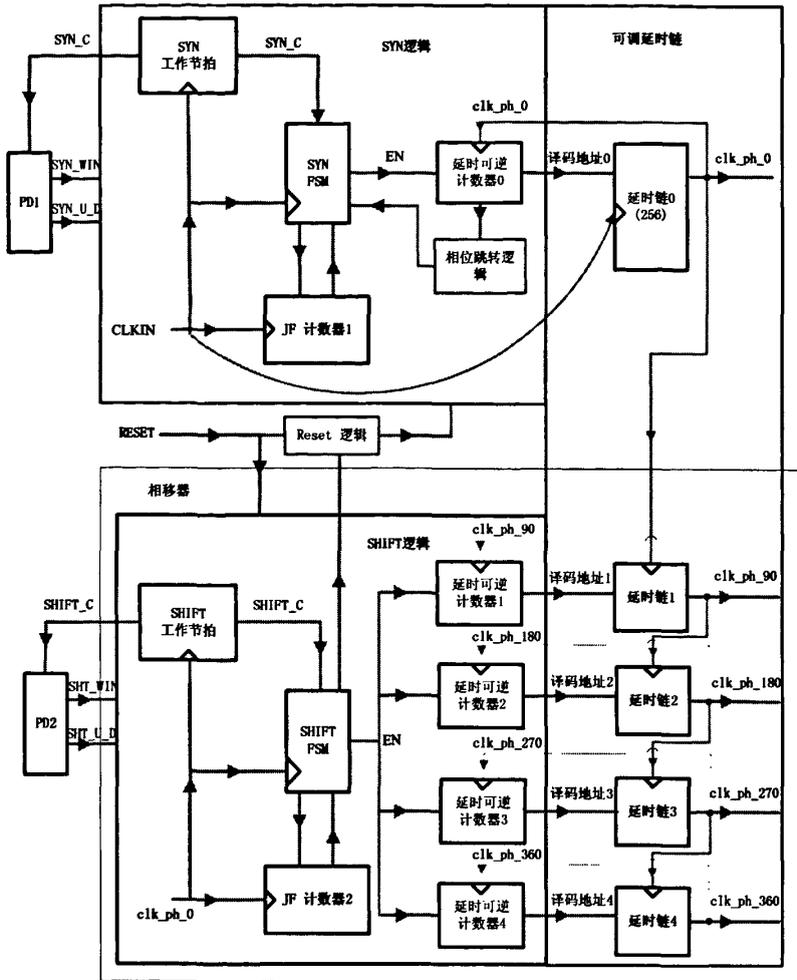


图 3-2 DLL 数字控制逻辑和可调延时链

从图 3-2 可见，可调延时链共 5 条，即一条主可调延时链（延时链 0，256 个延时单元），四条子可调延时链（延时链 1-4，各 128 个延时单元）。如图 3-2 所示，四条子延时链，SHIFT 逻辑和一个鉴相器（PD2）构成相移器。SHIFT 阶段，相移器工作。相移器采集第一级子延时链的输入时钟  $clk\_ph\_0$  和最后一级延时链的输出时钟  $clk\_ph\_360$ ，根据鉴相结果同步调整四条子可调延时链的延时，直至将  $clk\_ph\_0$  和  $clk\_ph\_360$  调整至同步。使得经过相移器的时钟延时是一个周期，从而使得时钟经过相移器中的每个子延时链的输出时钟相移  $90^\circ$ ，对应图 3-2 中分别为  $clk\_ph\_0$ ， $clk\_ph\_90$ ， $clk\_ph\_180$ ， $clk\_ph\_270$ ， $clk\_ph\_360$ ，这些相移的时钟可以根据实际的需要由时钟生成模块产生所需要分频（CLKDV），倍频（CLK2X），或移相时钟作为输出时钟，关于分频和倍频电路，参见 3.6 时钟生成模块设计一节。SYN 逻辑用于控制将反馈时钟和输入时钟调整至同步。

整个 SHIFT 阶段和 SYN 阶段都是在各自的控制逻辑模块控制下工作的，

以一定的工作节拍实施调整(工作节拍信号 SHIFT\_C,SYN\_C)，如图 3-3。

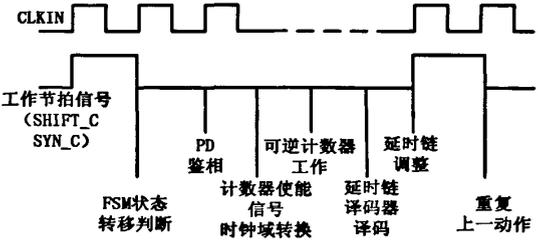


图 3-3 DLL 工作节拍流程

图 3-2 中工作节拍模块生成工作节拍信号 (SHIFT\_C,SYN\_C)。在工作节拍下，状态机处于某一状态，则根据状态的调整要求依次进行如下操作:鉴相，判断出输入时钟和反馈时钟的相位关系，超前/滞后 (SHT\_U\_D, SYN\_U\_D)，同时还可以指示两时钟是否进入锁定窗(SHT\_WIN, SYN\_WIN)，见图 3-1。锁定窗概念见 2.3.2 DLL 模块化结构及工作原理。鉴相器将这些信息送入控制逻辑模块，在 SHIFT 阶段四条延时链对应各自的可逆计数器，负责控制延时链加/减延时单元，各计数器工作在自己的时钟域中，如图 3-2。根据鉴相的结果和所处的状态机状态，计数器进行计数，计数结果作为延时链的译码地址，最后延时链经过地址译码增加/减少一个延时单元，完成一次工作节拍调节，继而继续进行下一次调整。直到状态机进入锁定状态为止。SYN 阶段工作方式类似，但只对主延时链进行调整。实现 DLL 锁定，同步建立需满足公式：

$$D_{SYN} + SKEW = mult(P) \quad (3-1)$$

- \* $D_{SYN}$  : 主延时链可以提供的延时
- \*SKEW 为时钟偏斜
- \* $mult(P)$ 为整数个输入时钟周期

状态机设计如下：

SHIFT: 复位状态 (000)，锁定窗检测状态 (101)，锁定窗内滞后检测状态(100)，锁定窗内超前检测状态(110)，释放 SYN 状态(111)，锁定状态(011)

RESET 后进入复位状态，所有计数器清零，当工作节拍信号使能时进入锁定窗检测状态，此时处于粗调使能阶段，计数器进行“+”状态，延时链不断加入延时单元使得 CLKFB 上升沿“后移”。当 CLKIN 和 CLKFB 进入锁定窗后 (CLKFB 仍超前于 CLKIN)，状态进入锁定窗内滞后检测状态，此时计数器将根据鉴相器的超前/滞后信号指示计数器工作，直至将反馈时钟信号初次调整至滞后于 CLKIN，此时进入锁定窗内超前检测状态，微调使能，指示计数器再次将 CLKFB 微调成超前于 CLKIN，则完成 SHIFT 阶段调整。等待 SYN 状态机工作，和 SYN 状态机一起进入锁定状态。

SYN: 复位状态 (000)，锁定窗检测状态 (001)，相位跳转状态 (111)，

锁定窗内滞后检测状态 (100), 锁定窗内超前检测状态 (110), 锁定状态 (011)

除相位跳转状态外, 其他几个状态同 SHIFT 基本相同。相位跳转状态是当 256 延时链所有延时单元用尽后仍然没有将 CLKIN 和 CLKFB 调整至同步, 则借用四个 128 延时链的时钟信号的相位作为反馈时钟延时链, 通过图 3-2 中的相位跳转逻辑实现。相位将立即变成  $(n + 90^\circ)$ , 相当于借用了四个 128 延时链的延时单元。则式 3-1 将变为

$$D_{\text{SYN}} + D_{\text{SHIFT}} + \text{SKEW} = \text{mult} (P) \quad (3-2)$$

\* $D_{\text{SYN}}$  : 256 延时链提供的延时

\* $D_{\text{SHIFT}}$ : 128 延时链提供的延时 (相位跳转可用延时)

可以将上式改写成,

$$D_{\text{INIT}_N} + D_{\text{SHIFT}_N} + \text{SKEW} = \text{mult}(P) \quad (3-3)$$

\* $D_{\text{INIT}_N}$  为选取相应相位时钟作为反馈时钟所对应的初始延时值 ID

$D_{\text{SHIFT}_N} = N * (P/4)$ ,  $N=0, 1, 2, 3$  则

$$\text{则 } D_{\text{INIT}_N} = \text{mult}(P) - \text{SKEW} - N * (P/4) \quad (3-4)$$

\*计算时保持两端都为正值。

SHIFT 和 SYN 的状态机设计遵循 DLL 原理阐述中的图 2-11 状态机控制器原理示意图。SHIFT 是时钟相移器的调整逻辑控制, SYN 是对时钟同步调整的逻辑控制, 锁定窗检测状态 (001), 相位跳转状态 (111), 锁定窗内滞后检测状态 (100), 锁定窗内超前检测状态 (110) 属于锁定获取阶段控制; 锁定状态 (011) 以及锁定后的抗抖动调整设计属于锁定保持阶段控制。

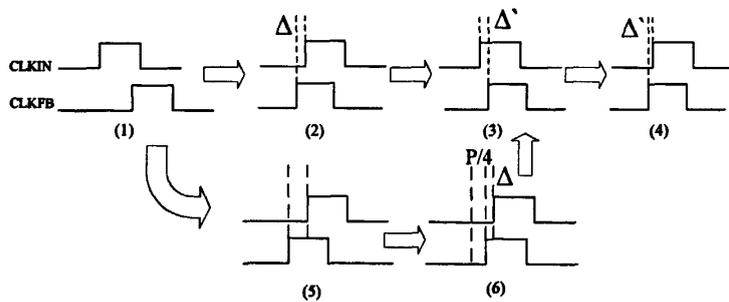


图 3-4 DLL 状态跳转与时钟调整

如图 3-4 所示,  $\Delta$  表示参考时钟和反馈时钟进入锁定窗,  $\Delta'$  表示锁定窗内调整相位值。(1) 为复位状态, (2) 为锁定窗检测状态, (3) 为锁定窗内滞后检测状态, (4) 为锁定窗内超前检测状态, (5) 为主延时链延时单元用尽后仍然无法同步两时钟, (6) 为相位跳转状态, 经过借用了  $90^\circ$  的相位 (1/4 周期)。SHIFT 阶段 (1) =>(2)=>(3)=>(4), 最终跳转到锁定状态。SYN 阶段当需要相位跳转时, 经过 (1) =>(5)=>(6)=>(3)=>(4), 否则同 SHIFT 阶段相同。通过状

态机的调整，使得反馈时钟和参考时钟同步，DLL 完成锁定。图 3-5 为状态机代码综合后状态转移图。

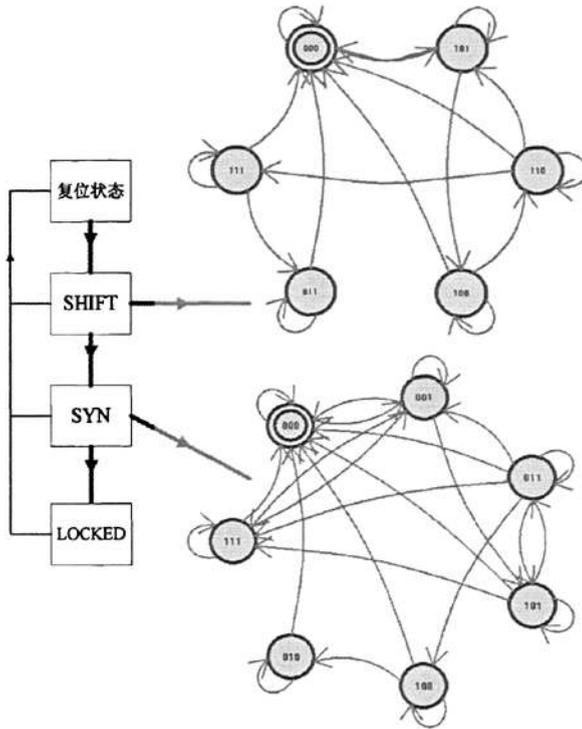


图 3-5 状态机状态转移图

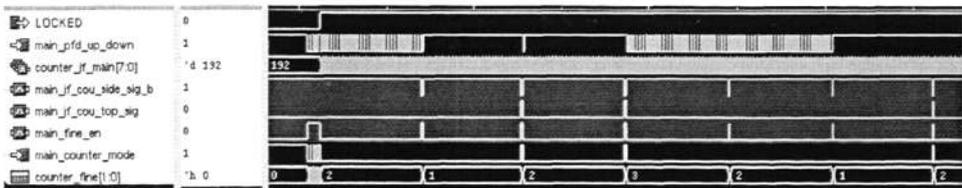


图 3-6 锁定后 COUNTER\_JF 调整

抗抖动模块设计：抗抖动模块为图 3-2 中的两个 JF 计数器及配合逻辑构成。如图 3-6，波形为 SYN 部分的 JF 计数器为例。LOCKED=1 后，由于设定的 JF 值为 C0('d192)，COUNTER\_JF 受控于 main\_pfd\_up\_down,也就是说根据 SYN 过程的超前/滞后信号控制，JF 计数器，按照给定的 JF 值进行加/减，在完成 JF 值的加减后，发出指示信号，指示微调延时单元计数器，进行加减操作，从而加入或减少微调延时单元值。例如，如图 3-6，JF 值为 192 ('b11000000)，锁定后，同样是在工作节拍信号的控制下，首先判断 JF 计数器计数值是否等于 JF 值，如果是，则按位取反，得到 63 ('b00111111)，然后根据 main\_pfd\_up\_down

动作，如波形图 3-6，main\_pfd\_up\_down=0（采样值，反馈时钟滞后于输入时钟），则 JF 计数器做减操作，当减到 0 时，发出指示信号，指示微调延时计数器减一，则微调延时单元减少，使得反馈时钟超前于输入时钟，main\_pfd\_up\_down=1（采样值）。然后 JF 计数值按位取反 0->192,从 JF 值开始做加操作，直至加至 255，然后同样发出指示信号，指示微调延时计数器加一，即增加微调延时单元，经过这样的加/减，锁定时钟会在很小的范围内增减相位。以此方式，反复的微调，可以在时钟抖动发生的情况下，按照 JF 值的设定，周期性的对锁定后的时钟进行微调。抵消时钟抖动的作用，保持锁定时钟的稳定输出。

SHIFT 部分的 JF 计数器的工作机制同 SYN 相同，只是针对的时钟信号是 clk\_phase90, clk\_phase180, clk\_phase270, clk\_phase360。

如下图 3-7 JF 流程示意图所示，锁定后的抗抖动调整，内抗抖动调整：SHIFT JF 计数器根据鉴相器的鉴相结果（up, down），按照 JF 设定值，开始或加或减的操作，形成一个计数周期，比如 JF=192 则形成了一个 64 的计数周期（如上文所述）。在计数到达计数周期后，JF 计数器发出指示信号，指示状态机控制延时计数器，增加或减少延时链的延时单元数，从而调整了 SHIFT 的反馈时钟（clk\_phase360），从而进入下一轮鉴相调整过程。由于这些时钟信号和鉴相过程都是在内部完成的，所以称为内部抗抖动调整，如图 3-7 inside 箭头。

外部抗抖动调整：即 SYN 部分的抗抖动调整，调整原理相同，同样是根据 JF 设定值，进行周期性的调整。SYN 部分的抗抖动调整是直接针对输出时钟的，所以称作外部抗抖动调整，如图 3-7 中 outside 箭头。

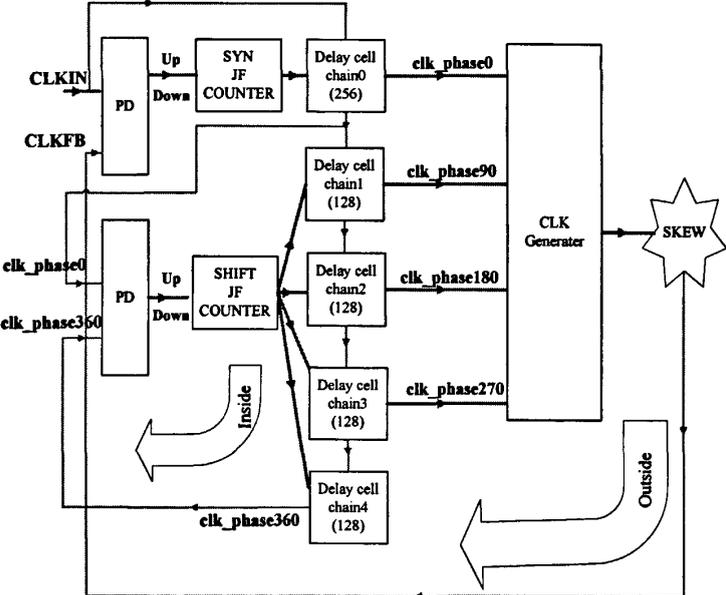


图 3-7 JF 逻辑结构示意图

在 0.25um CMOS 工艺下，将以上数字控制部分的代码进行综合，经过时序优化后，如图 3-8 为综合后电路框图。

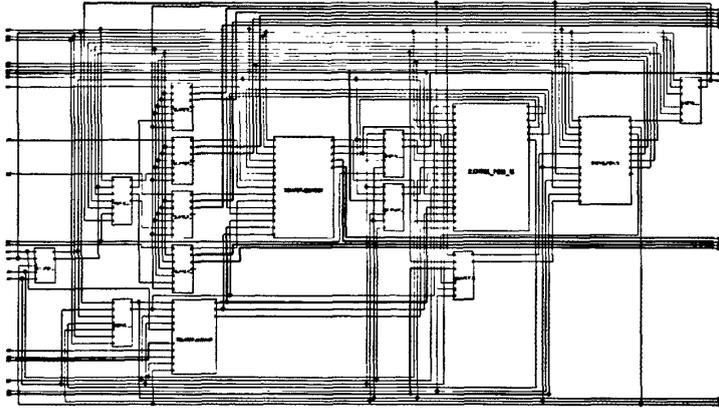


图 3-8 数字控制部分综合后电路框图

### 3.3 鉴相器(Phase Detector)设计

#### 3.3.1 鉴相器结构

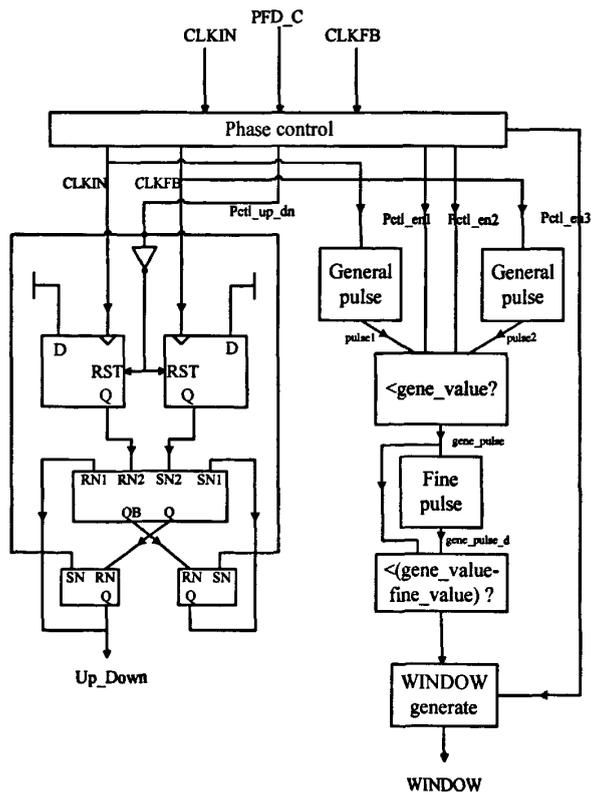


图 3-9 鉴相器结构框图

如图 3-9 为鉴相器结构框图, 左边为超前/滞后信号 UP\_DOWN 的电路结构, 右边是锁定窗信号 window, 表明输入时钟和反馈时钟已经进入锁定窗, 指示数字控制逻辑进入相应的状态, 进行进一步调整。PFD\_C 是工作节拍信号, 由工作节拍模块 (图 3-2) 生成。phase\_ctl 用于产生与 PFD\_C 相位相差 0.5, 1, 1.5 个输入时钟周期的信号。以及与 PFD\_C 相位相差 0.5, 1, 1.5 个反馈时钟周期的信号, 这些信号用于配合输入时钟信号和反馈时钟信号用于在 PFD\_C 的固定相位处鉴相。这里规定, 在 PFD\_C 高电平后的第二个时钟周期开始鉴相。由于鉴相是对时钟上升沿进行的, 所以 Pctl\_up\_dn, 用于在 PFD\_C 高电平之后的第一个时钟周期 (无论是输入时钟还是反馈时钟), 对鉴相电路进行清零, 并置位输出的 UP\_DOWN 信号, 为第二个周期进行的鉴相做准备。Pctl\_en1, Pctl\_en2, Pctl\_en3 用于配合输入时钟和反馈时钟信号, 生成粗鉴 (general\_pulse) 和微鉴 (fine\_pulse) 脉冲信号, 并通过两级判断电路, 即分别判断两时钟之间的相位差是否小于某个粗鉴值 (比如 0.8ns), 如果是, 那么继续判断两时钟之间的相位差是否小于某个细鉴值 (比如 0.6ns), 如果是那么 window 信号生成。

### 3.3.2 鉴相器电路设计

基于全定制流程完成的 Phase Detector (简称 PD) 设计。PD 包括两个功能, 1 鉴相输入时钟和反馈时钟的超前/滞后关系。2 判断输入时钟和反馈时钟上升沿间的间距是否小于设定值 (即判断反馈时钟是否进入锁定窗)。鉴相器电路设计如图 3-10。

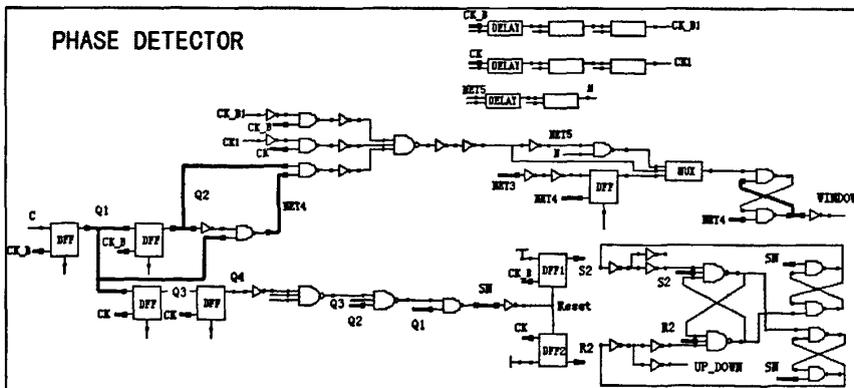


图 3-10 鉴相器电路结构图

鉴相器具体工作机制如下:

(1) 超前/滞后指示信号 UP/DOWN: 如图 3-10 下半部分为超前/滞后鉴相电路, 生成超前/滞后指示信号 UP\_DOWN。

C 为 CLK\_REF 生成的占空比为 1:5 的工作节拍信号。q1, q2 为同

CLK\_REF 固定相位关系的信号，q3，q4 为同 CLK\_FB 固定相位关系的信号。

具体关系为，q1 为 C 后延一个 CLK\_REF 时钟周期（同步于 CLK\_REF 上升沿），q2 为 C 后延 1.5 个 CLK\_REF 时钟周期（同步于 CLK\_REF 下降沿），q3 为 q1 时钟由 CLK\_FB 下降沿寄存（同步于 CLK\_FB 下降沿），q4 为 q3 时钟由 CLK\_FB 上升沿寄存。

$Reset = q1 \& \!(q2 \& q3)$ , when en\_main\_b = 0, 即 CLK0 作为反馈时钟

$Reset = q1 \& \!(q2 \& q3 \& q4)$ , when en\_main\_b = 1, locked=1 即 CLK2x 作为反馈时钟

$SN = !reset$

图 3-11 是 CLK0 作为反馈时钟，CLK\_FB 滞后于 CLK\_REF 的情况。鉴相是在 C 的高电平后的第二个 CLK\_REF 时钟周期的上升沿进行的。如图 3-11，reset 信号确保在 C 后的第二个 CLK\_REF 上升沿到来前，即 C 后第一个时钟周期的下降沿（可能是 CLK\_REF 或者是 CLK\_FB 的），清零 s2，r2，然后迎接上升沿的到来，进行鉴相。注意为保证鉴相结果稳定，电路采用的两级 RS 结构。图 3-12 为 CLK\_FB 超前于 CLK\_REF 的情况。

在鉴相准备期间（reset=1），SN=0，即保证第一级 RS 的 S1 和 R1=1，第一级 RS 只受 S2,R2 影响。

DLL 工作中可以选取 CLK0（0° 相移时钟）或 CLK2X（0° 相移，倍频时钟）作为反馈时钟。当 CLK2X 作为反馈时钟的时候，锁定前 q4 没有作用，鉴相过程同 CLK0 为反馈时钟时相同。锁定后 q4（跟踪 CLK2X 反馈时钟的第二个上升沿）有效，reset 信号确保在输入时钟的 C 后第二个上升沿和反馈时钟的第三个上升沿前，清零判断电路。

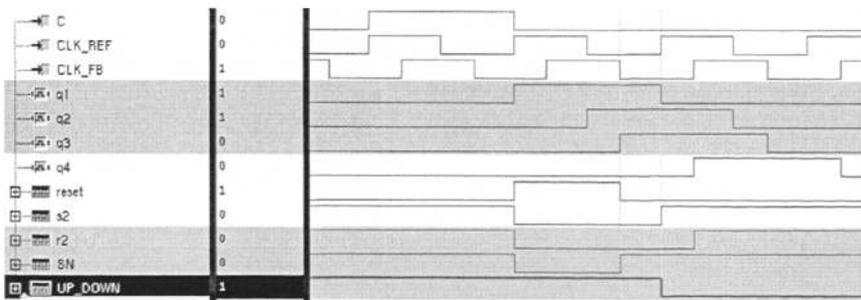


图 3-11 鉴相器超前/滞后信号(CLK\_FB 滞后于 CLK\_REF)

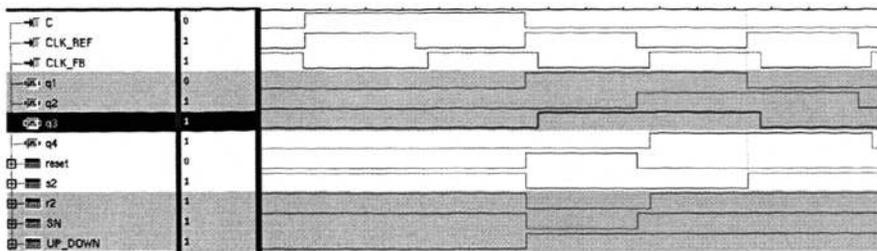


图 3-12 鉴相器超前/滞后信号(CLK\_FB 超前于 CLK\_REF)

(2) 锁定窗指示信号 Window 生成:

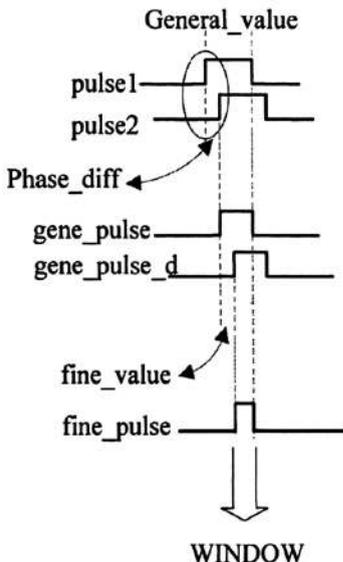


图 3-13 WINDOW 生成信号关系

信号关系建模，如图 3-13。首先进行粗鉴，通过 pulse1 和 pulse2（两信号分别跟踪输入时钟和反馈时钟的上升沿）的信号关系生成 gene\_pulse，即粗鉴脉冲信号，表示两个时钟上升沿之间的相位差小于 0.8ns。将 gene\_pulse 延时 0.2ns 得到 gene\_pulse\_d 信号,通过 gene\_pulse 和 gene\_pulse\_d 的相位关系生成细鉴脉冲信号 fine\_pulse。表示两时钟间的相位差小于 0.6ns，则生成 window 信号，表示两个时钟的相位关系已经进入锁定窗了。

Window 信号生成波形关系图，如图 3-14。CLK\_REF 同步于 REF0.8(脉宽为 0.8ns 的脉冲)，CLK\_FB 同步于 FB0.8(脉宽为 0.8ns 的脉冲)。ENABLE 是鉴相许可信号，是 C 信号延迟半 CLK\_REF 的版本（因为 C 高电平后第二个周期进行鉴相）。gene\_pd 是粗鉴信号， $gene\_pd = !(REF0.8 \& FB0.8 \& ENABLE)$ ，也就是说当两时钟的相位差小于 0.8ns 时，粗鉴脉冲有效。fine\_pd 是细鉴信号，

当两时钟的相差小于 0.6ns 时，生成细鉴脉冲。由细鉴脉冲生成 window。

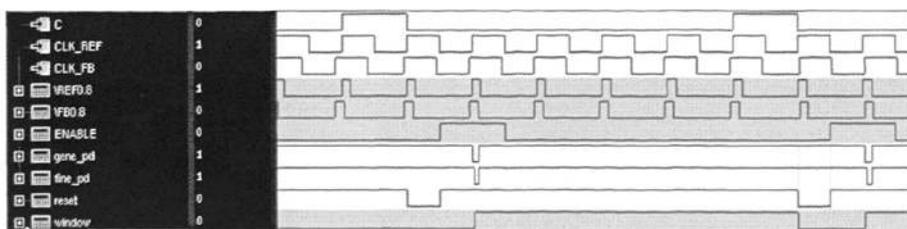


图 3-14 WINDOW 信号波形关系图

PD 在工作节拍信号下，鉴别输入时钟和反馈时钟的相位，生成超前/滞后信号和 window 信号，这两个信号输入到状态机中，配合其他信号控制状态机状态跳转。DLL 是六个周期进行一次相位调整的。鉴相在第二个周期进行。鉴相的结果在剩余的工作节拍周期内保持，直到下一个工作节拍的第一周期，根据这两个信号判断状态跳转，然后复位这两个信号。再进入新一轮的相位判断。

### 3.4 延时链系统设计

#### 3.4.1 基本延时单元

延时链<sup>[20]</sup>的基本延时单元由两个 CMOS 反相器串联构成，如图 3-15。由于 CMOS 晶体管可以模拟成 RC 电路的电阻，晶体管驱动的信号线的杂散电容可作为 RC 电路的电容，电容的充放电时间形成 CMOS 晶体管反相器的传播时延。两个反相器的串联在逻辑上是可以消去的。它不改变信号，但产生了一个小的时延，该时延等于两个 CMOS 晶体管反相器传播时延之和。控制器可以通过调整反相器的 MOS 管的数目来调节延迟时间。基本延时单元的延时是调节时钟延时的基本刻度。延时链的最大延时值可以通过计算基本延时单元的数目来设定。

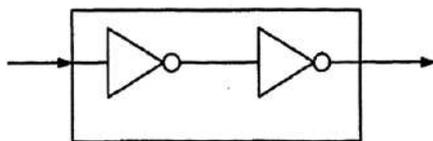


图 3-15 基本延时单元

#### 3.4.2 延时链电路设计

延时链系统<sup>[21]</sup>包括五条延时链，工作机制是相同的，每条延时链与逻辑控制逻辑中的可逆计数器及译码电路组成延时链自调整系统，如图 3-16，在控制逻辑的作用下调整输入时钟，相应的减少/增加延时插入值。

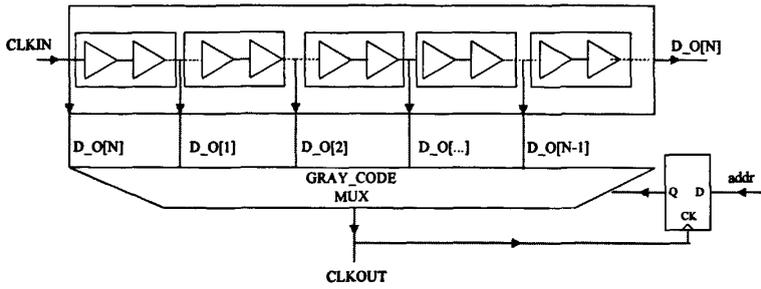


图 3-16 延时链功能结构

具体工作过程：可逆计数器的计数值（表征控制部分发出的增加/减少相位命令）作为地址输入到 2-4GRAY 码译码器，经过寄存器寄存后，将译码结果输出到延时链的 MUX，控制延时单元的插入。寄存器的时钟由延时链的输出端提供，目的是为消除毛刺。因为用调整后的时钟控制译码结果的输入，不会生成毛刺。如果用正在调整的时钟控制译码则容易出现“暂态”。

数字控制逻辑输出的延时单元加/减指示信号,即延时链的地址信号是 GRAY 码，主要是出于低功耗的考虑。地址信号经过 2-4GRAY 码译码器将输入的两位 GRAY 码译成四位的独热码，用独热码控制开关树的开启或关断。

延时链包括粗调和微调两部分电路。粗调延时单元电路工作过程如图 3-16 所示以及上文的介绍。微调延时单元电路<sup>[22]</sup>同样是由译码电路译码得到的独热码。进行开关路径选择控制。延时工作机制相同，在粗调完成后微调译码及调整开始。

延时链自调整系统，如图 3-17。COUNTER\_DEALY 表示数字控制逻辑中的可逆计数器，在鉴相器的超前/滞后指示信号和锁定窗信号的控制下，通过控制逻辑控制计数器的加减，相应的输出计数器的计数值，作为 GRAY\_DECODER 的地址信号，经过译码器译码得到延时链独热码地址信号，即选择延时链的延时单元个数，控制时钟树选择相应的延时路径，从而完成对目标时钟的延时操作。调整后的时钟反馈到数字控制逻辑中的相应可逆计数器，如此往复进行时钟调整。SHIFT 和 SYN 阶段的延时链自调整过程相同。

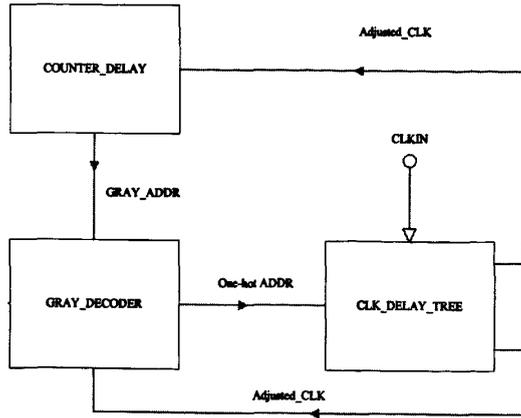


图 3-17 延时链自调整系统

如图 3-18 所示，SHIFT 阶段，四条子延时链在各自的时钟域内完成时钟调整。通过 SHIFT 状态机控制，增加/减少延时链延时单元，从而逐渐将四个时钟 clk0, clk90, clk180, clk270, clk360 调节成相位相差 90 度的时钟。

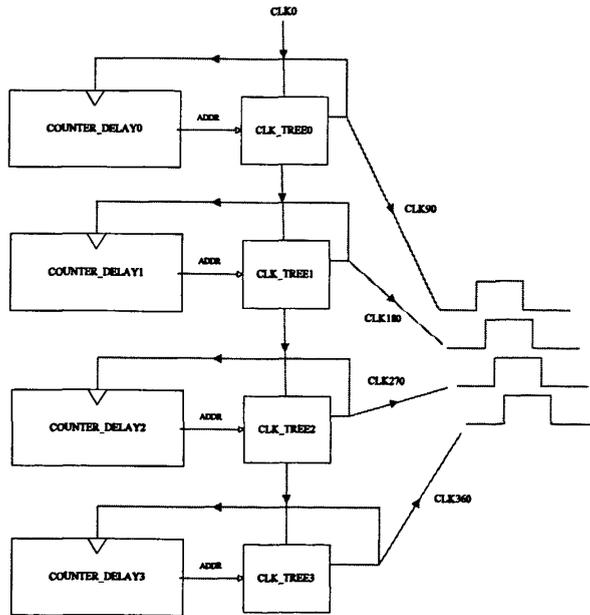


图 3-18 子延时链 SHIFT 调节示意图

如图 3-2 DLL 数字控制逻辑和可调延时链，延时链系统中五条延时链串联起来。SHIFT 阶段主延时链（256 个延时单元）不插入延时单元，时钟以“零”延时通过主延时链！四条子延时链根据地址译码值，增加/减少延时单元，高频模式，1, 3 子延时链工作，0, 2 子延时链短接。延时链的译码是在各自的时钟域内完成的，彼此不受影响。具体的工作流程已在逻辑控制一节中详细论述。

五条延时链可以形成环震，通过外部控制逻辑信号选择从哪里返回环震的末端，比如可以从 clk90 度返回，这时环震的长度最大值是  $256 + 128$  个延时单元，如果从 clk180 度返回，这时环震的长度最大值是  $256 + 2 \cdot 128$  个延时单元，依此类推。DLL 环震设计配置可以在 FPGA 片内形成时钟源方便芯片功能测试，具体应用见第五章 DLL 应用。

延时链的延时单元的精度，很大程度上制约了 DLL 的调整精度，所以为确保延时单元的延时值稳定，DLL 内加入了一个 LDO 对各延时链的延时单元供电。

### 3.5 LDO 设计

DLL 延时单元是单独供电的，由 LDO<sup>[23] [24]</sup> 将片内 2.5V 的电源电压转化成局部的 2.0V 电源，目的是消除 FPGA 全局电源不稳定和噪声对延时单元的影响，保证 DLL 输出时钟有较好的抖动特性。FPGA 中的 4 个 DLL 分布在芯片的 4 个角上（版图布局），各使用 1 个 LDO 单独供电。LDO 由经典的两级运放驱动一个比较大的 PMOS 管，使用米勒电容和调零电阻进行频率补偿，配合输出端挂接大量的 MOS 电容，可以提供几十个 mA 的电流。

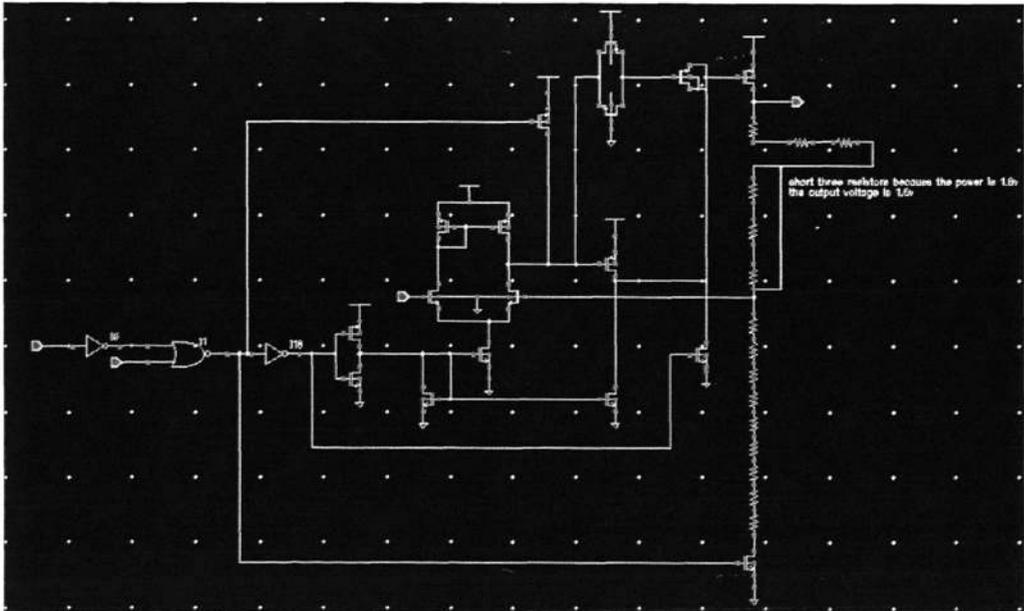


图 3-19 LDO 电路结构

图 3-19 是 DLL 中的 LDO，采用差分输入+两级共源结构，在第一级和第三级之间采用零点补偿（RC 串联），输出电压为 2v（参考电压为 1.2v，负载稳压电容在外部，采用三级结构是在较大的 ID 下，保证 OP 有足够的 Gain，ID 较大，OP 受外界影响较小）。



CLK2X 生成电路，如图 3-23:

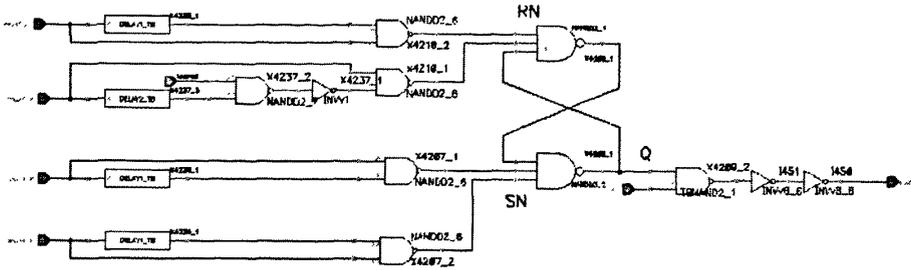


图 3-23 CLK2X 生成电路

CLK2X 同样是由相位不同的时钟作为输入，用脉冲生成的原理产生四个脉冲信号，经过 RS 电路采样，生成时钟。由于生成的时钟只和输入时钟的边沿有关，所以当输入时钟的相位关系固定后，生成时钟可以稳定的输出。

图 3-24 为分频电路系统结构框图

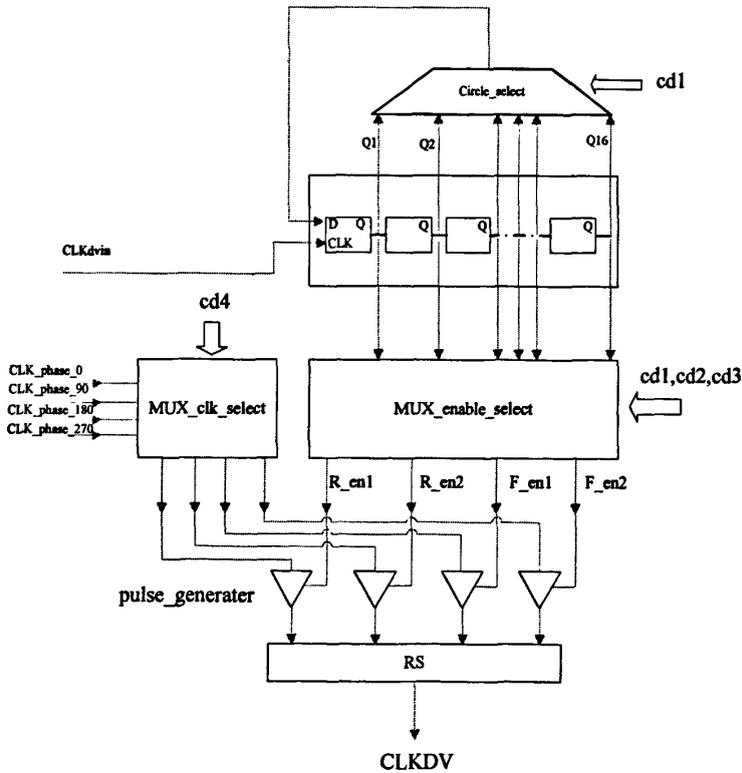


图 3-24 分频电路结构图

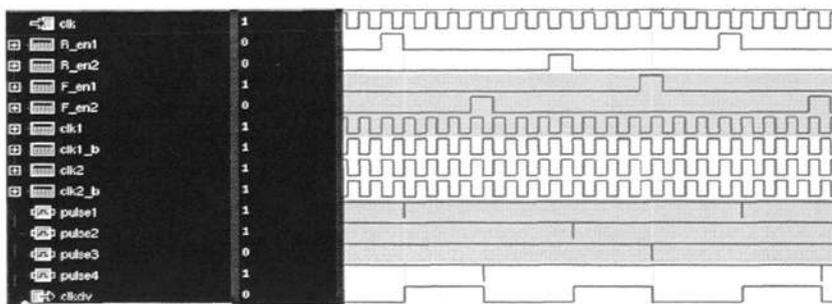


图 3-25 分频时钟信号模拟 (3.5 分频)

根据配置点配置, 如图 3-25 3.5 分频。clk1 配合 R\_en1 生成 pulse1, 触发生成 clkdv 的第一个上升沿; clk1\_b (clk1\_b 与 clk1 相位相差 90 度) 配合 R\_en2 生成 pulse2, 触发生成 clkdv 的第二个上升沿; clk2 配合 F\_en2 生成 pulse4, 触发生成 clkdv 的第一个下降沿; clk2\_b 配合 F\_en1 生成 pulse3, 触发生成 clkdv 的第二个下降沿。

R\_en1, R\_en2, F\_en1, F\_en2 的相位关系是通过图 3-25 中 MUX\_enable\_select 开关选择模块生成的。MUX\_enable\_select 根据 Q1...Q16 的输入时钟信号, 依据配置点选择 Q1...Q16 或者 Q1...Q16 的半周期延时版本, 作为使能输出信号 R\_en1, R\_en2, F\_en1, F\_en。具体的配置点设定信息如下:

cd1: 决定分频周期数, cd1[3:1] 决定 CLKDV 上升沿使能信号 R\_en2, R\_en2 出现在 cd1 设定的周期的半周期下降沿位置。(R\_en1 = Q1)

cd2: 决定 CLKDV 下降沿使能信号 F\_en2

cd3: 决定 CLKDV 下降沿使能信号 F\_en1

cd4: 决定脉冲生成电路的输入时钟信号, 同时配合控制信号实现相位跳转。

四个脉冲生成电路的输入信号两两成为一对相位相差 180 度的时钟信号, 例如 clk\_phase0 同 clk\_phase180, clk\_phase90 同 clk\_phase270。如图 3-25 clk1, clk1\_b, clk2, clk2\_b。

四个使能信号配合 clk1, clk1\_b, clk2, clk2\_b 便可以生成任意 1/4 分频。

### 3.7 OSDLL 架构设计

one-shot 逻辑<sup>[26]</sup>的主要结构主要如图 3-26。START 信号同 CLKFB 同步, STOP 信号同 CLKIN 同步, START (恒“1”)信号送入延时链的输入端, 每个计算单元的输出端作为译码电路的输入信号, STOP 为译码器的采样信号。通过译码电路计算出两者之间的延时单元数目。如图 3-27 将计算好的延时单元的数目 DELAY\_NUM 送入数字控制模块(DIGITAL)的可逆计数器中。启动 SYN 阶段, 开始工作。

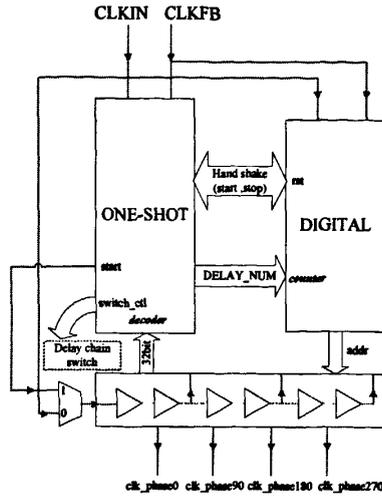


图 3-26 One-shot 逻辑与数字控制逻辑关系图

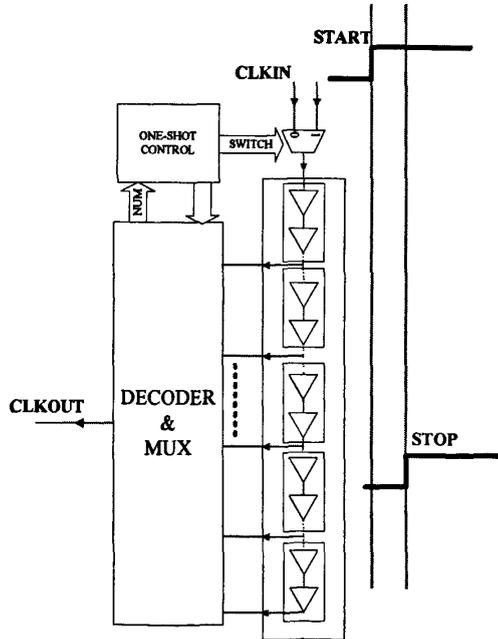


图 3-27 One-shot 计算过程

如图 3-26，根据前文介绍的 one-shot 工作原理，经过 SHIFT 阶段后 DLL 进入 one-shot 计算阶段，此时主延时链用于进行对反馈时钟和输入时钟的粗略延时值进行计算，如图 3-27。START 和 STOP 分别跟踪反馈时钟和输入时钟。首先清零延时链，使得各延时单元的输出级输出“0”，当 START=1 时，一个恒“1”值在主延时链中传播，当 STOP=1 时，译码电路对延时链的各级延时单元抽头出的一组信号（以 8 个基本延时单元为一个译码输出单位）进行译码，即如图 3-27，译码一组“1111.. 10000”数据，计算出“1”的个数，则可以

得到两个时钟边沿之间的延时值粗略的等于多少个计算延时单元 NUM。将计算出的数值付给数字控制逻辑中的可逆计数器作为初始计数值，则大部分的延时可以在计数器初始计数时已经消除了，大大的提高了锁定的速度，同时不影响控制逻辑原有的调控机制。当完成 one-shot 计算过程后，如图 3-26，通过握手信号通知数字控制逻辑进入 SYN 阶段，此时主延时链转为用作输入时钟延时调节。

OSDLL 结构只是在原 DLL 架构基础上增加了部分控制逻辑和时钟选择开关，由于复用了主延时链作为 one-shot 计算延时链，节省了很大一部分电路资源，而同时可以很大程度的提升锁定速度。

### 3.8 本章总结

本章介绍了 DLL 的整体结构设计,包括数字控制逻辑设计，鉴相器设计，可调延时链设计，以及时钟输出模块设计等。同时介绍了基于原有 DLL 结构实现的 OSDLL 结构设计。

## 第四章 DLL 功能与性能仿真及流片结果

### 4.1 FPGA 片内 DLL 整体参数仿真

基于全定制 FPGA 电路模型建立 DLL 仿真模型。如图 4-1 所示，为简化仿真过程选取 FPGA 电路中的一个 DLL 电路，加上选取固定的时钟网络和反馈网络（如图 4-1 中箭头所示）构成 DLL 同步调整的仿真系统。时钟网络的加入相当于 DLL 输出带上所有负载，则可以对寄生 RC 进行估算。

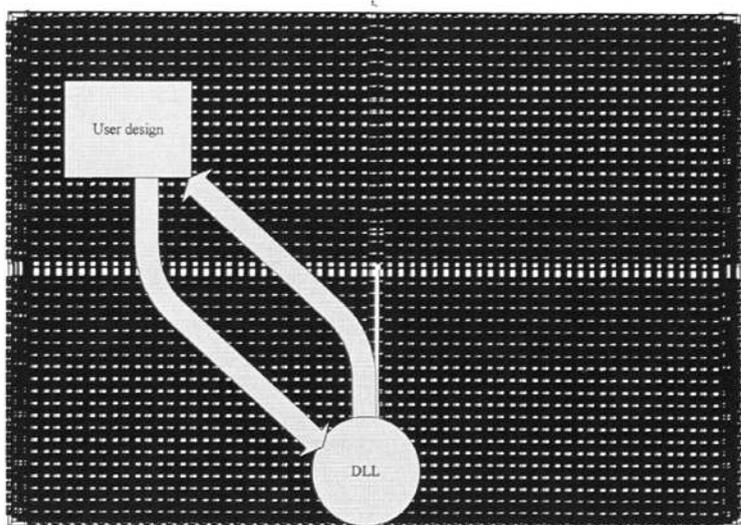


图 4-1 FPGA 整体仿真电路模型

图 4-1 是 DLL 的顶层电路图，箭头表示的线路包括全局缓冲器 GBUF 以及时钟网络。

根据图 4-1，抽象出如下模型：

如图 4-2，选取 FPGA 其中一个引脚 PAD，作为参考时钟输入，CLK\_IN 作为输入时钟进入 DLL 电路，DLL 输出时钟 CLK\_OUT 经过 GBUF 和全局时钟网络再返回 DLL。

FPGA 配置点依据以下要求：

DLL 按照默认值配置，目的是为了简化模型；

DLL 选择 PAD 输入时钟作为参考时钟，GBUF 选择 DLL 的 CLK0 作为输入时钟。

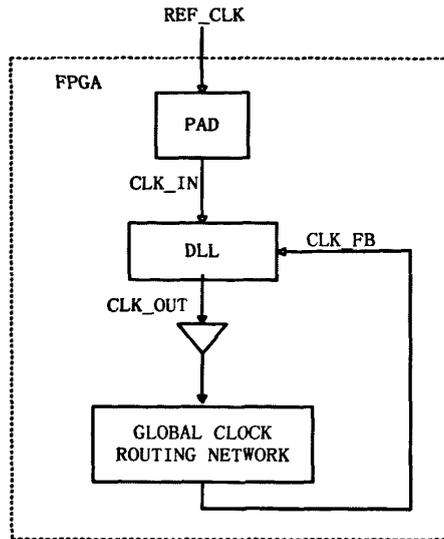


图 4- 2 DLL 模型仿真电路结构

### 1 RC 寄生模型

为了较为真实的模拟寄生参数，这里采用分布式的 RC 网络，即把计算出来的总电阻和总电容等分，如图 4-3。

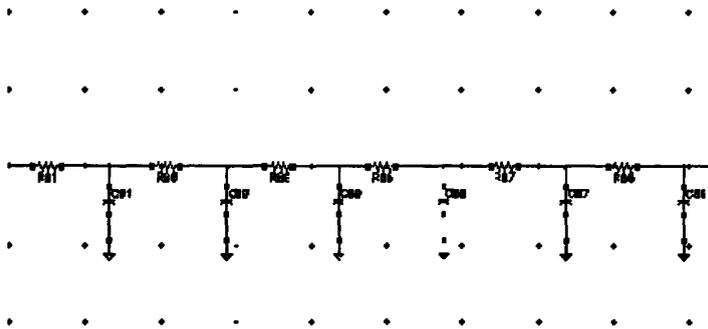


图 4- 3 分布式 RC 网络模型

### 2 仿真结果

VDD=Vdelay=2.5v(外加电源激励), Temp=27 , Model=TT, TGCLK=10ns, Th\_width=5ns(时钟高电平脉冲宽度)配置信号均为默认值, PAD 输入时钟, CLK0 输出作为 GBUF 输入。整体仿真结果见表 4-1 至 4-4:

表 4-1 高/低频模式下的可调时钟频率范围

	最大值	最小值	单位
锁定时钟频率（高频模式）	400	35	MHz
锁定时钟频率（低频模式）	200	12.5	MHz
输入时钟脉宽（高频模式）	--	1.5	ns
输入时钟脉宽（低频模式）	--	2	ns

表 4-2 高/低频模式下的可调时钟锁定时间

	输入时钟	高频		低频		单位
		最大值	最小值	最大值	最小值	
锁定时间	12.5-20M	--	--	300	--	$\mu s$
	20-30M	--	--	140	--	$\mu s$
	30-40M	75	--	100	--	$\mu s$
	40-50M	45	--	65	--	$\mu s$
	50-60M	50	--	35	--	$\mu s$
	>60M	30	--	20	--	$\mu s$

表 4-3 高/低频模式下的动态功耗

	最大值	最小值	单位
动态功耗（高频模式）	170	40	mA
动态功耗（低频模式）	130	30	mA
静态功耗	1.91	1.17	mA

表 4-4 仿真模型电路延时值

	PAD 延时时间		单位
	最大值	最小值	
反相器直通	530	310	ns
比较器直通 1 (VREF=0.75v)	700	400	ns
比较器直通 2 (VREF=0.75v)	660	390	ns
GBUF(全局缓冲器) 延时时间	0.74	0.39	ns

目前我们设计的 FPGA 芯片工作频率在 20-200MHz 之间。通过 FPGA 片内 DLL 的整体参数仿真结果显示, DLL 能够在 FPGA 的工作频率区间内正常工作。同时仿真结果显示 DLL 的锁定时间随着输入时钟频率的降低而逐渐升高, 这同 DLL 的调整机制有很大关系, OSDLL 设计将在锁定时间上对 DLL 加以改进。

#### 4.2 鉴相器参数仿真

基于全定制设计完成的 DLL 鉴相器, 具有配合数字控制逻辑工作节拍信号, 鉴别输入时钟和反馈时钟的超前滞后关系, 以及两时钟是否进入锁定窗的功能, 通过超前/滞后指示信号 UP\_DOWN 和锁定窗信号 WINDOW 来表示。基于鉴相器全定制电路设计结构, 对鉴相器进行 spice 参数仿真, UP\_DOWN 和 WINDOW 信号参数仿真结果如图 4-4, 图 4-5。

仿真条件: smic0.25um 工艺库, MODEL=TT, TEMP=27

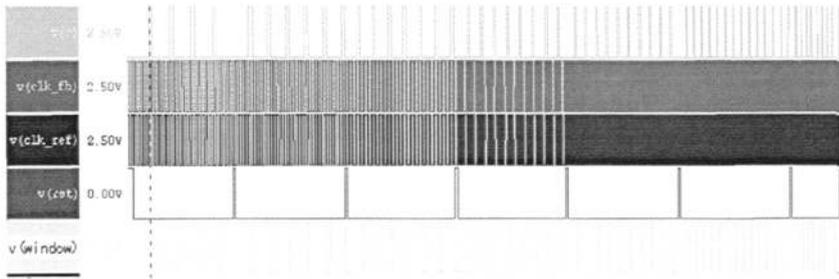


图 4-4 Window 信号频率扫描 (60MHz -> 500MHz 正常响应)

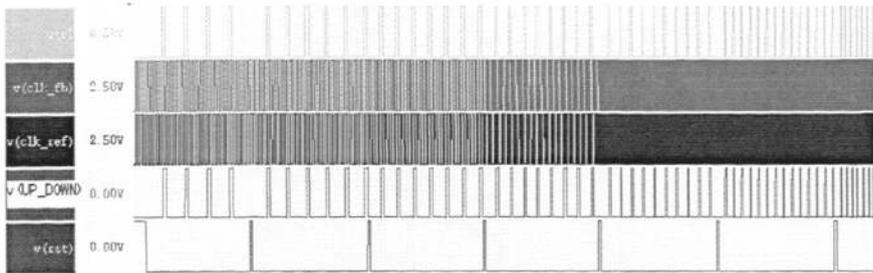


图 4-5 Up\_down 信号频率扫描 (60MHz -> 500MHz 正常响应)

鉴相器的参数仿真结果显示在一段较宽的时钟频段内, 鉴相器都能够正常响应。鉴相器的工作频率范围大于 DLL 的频率设定, 在 DLL 的工作频率段内, PD 能够匹配工作, 并且其较强的性能为 DLL 在高频锁定调整带来优势, 方便未来设计升级更高频率 DLL 版本。

#### 4.3 LDO 参数设计与仿真

仿真环境: MODEL=TT TEMP=27 VDD=2.5

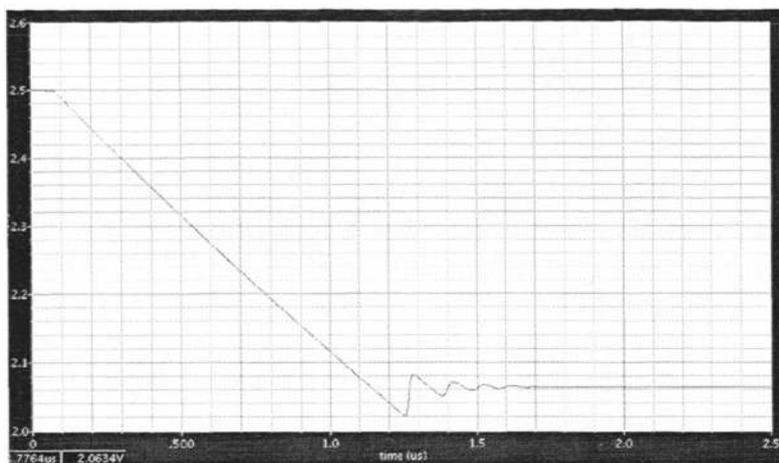


图 4- 6 LDO 输出电压 VOUT

如图 4-6 为 LDO 稳压过程仿真。LDO 稳定在 2v 左右。

驱动能力：在负载为 30mA 时候，电压大约下降 15mv；负载为 40mA 的时候，电压大约下降 0.1v。

静态功耗：1.7mA

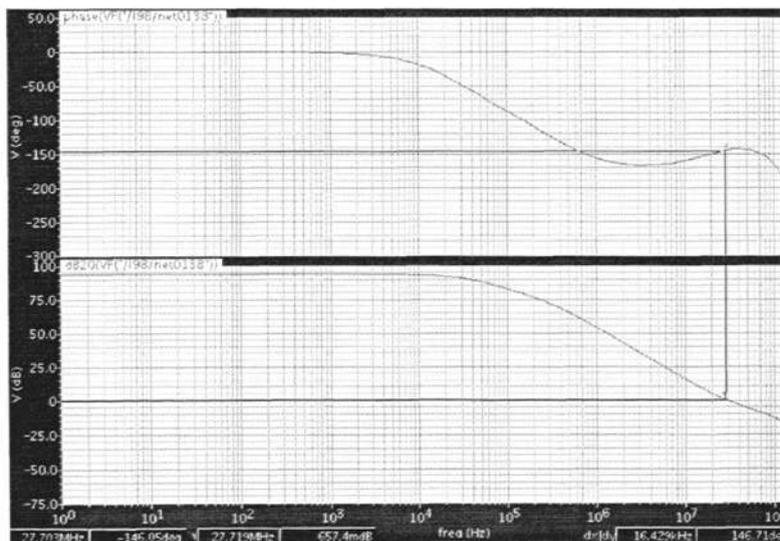


图 4- 7 LDO 交流仿真图

空载：dc gain=93.6dB，PM=34，如图 4-6 LDO 交流仿真图。其他仿真结果如下

1mA 负载：dc gain=92.1dB，PM=38

5mA 负载：dc gain=78.8dB，PM=43

10mA 负载：dc gain=70dB，PM=56

20mA 负载：dc gain=58.2dB，PM=92（均带上稳压电容）

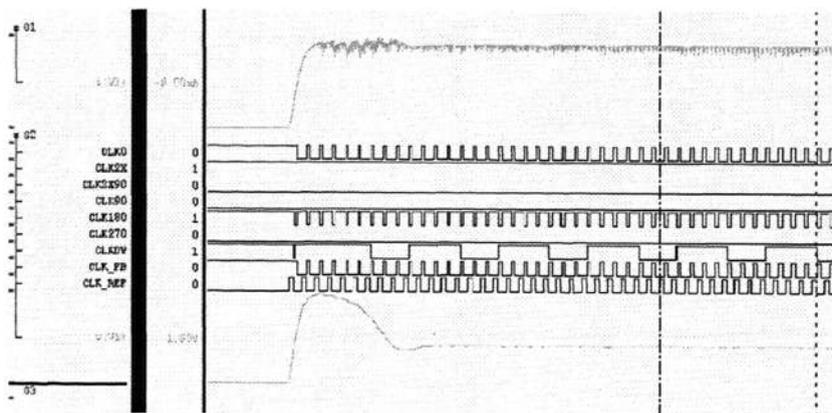


图 4-8 高频模式下 LDO 抖动影响

图 4-8 为 DLL 被配置成高频模式下，当输入时钟较高时的 LDO 输出 (V1) 仿真结果。

高频：负载大约是 10mA 左右，LDO 输出 V1 抖动基本上在 50mv 之内。

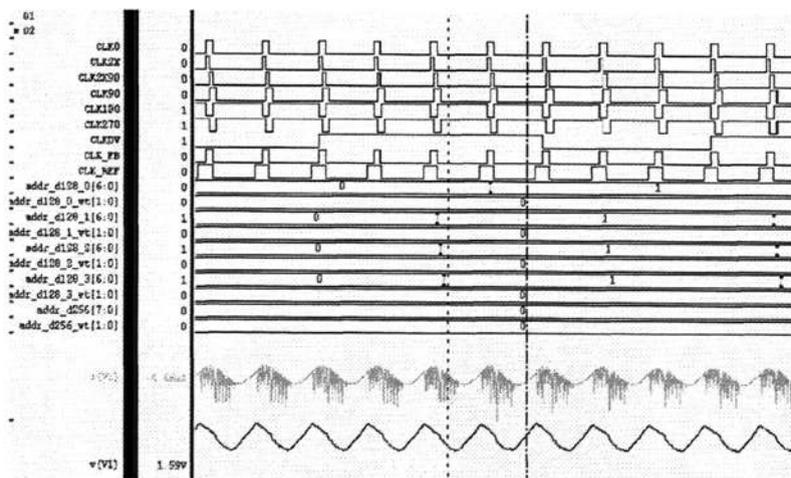


图 4-9 低频模式下 LDO 抖动影响

图 4-9 为 DLL 被配置成低频模式下，当输入时钟较低时的 LDO 输出 (V1) 仿真结果。

低频：负载大约是 5mA 左右，LDO 输出 V1 抖动基本上在 100mv 之内。

此两种情况下的仿真结果负载对 LDO 的稳定性不会造成较大影响。

#### 4.4 DLL 主要功能仿真

FPGA 电路的配置过程完成后便可以根据需要，对 DLL 的功能进行全面的验证。下面是根据实际各种配置情况对 DLL 进行验证。其中 DIV：是指分频系数；DUTY\_CORRECT：是占空比系数，当它为 TRUE 时，输出时钟占空比为

50%，当它为 false 时，输出时钟占空比保持与输入时钟相同。

DLL 需要进行的功能性验证包括如下几方面：DLL 的普通和低频模式下能否锁定，以及能否正确输出时钟 CK0, CK90, CK180, CK270, CK2X, CK2X180, DIV (1.5, 2, 2.5, 3, 4, 5, 8, 16)。

部分配置情况下的仿真波形，如图 4-10~图 4-20：其中 DUTY\_CORRECT=TURE 时，输出波形占空比为 1: 1，为 FALSE 时，输出波形与输入波形同样占空比。波形图中 CK\_IN 为输入时钟，CK\_B 是反馈时钟，LOCKED 信号为 DLL 的锁定信号，当 LOCKED 为高电平时，DLL 锁定，相反则未锁定。CK0 为输出零相移时钟，CK90 为输出 90 度相移时钟，CK180 为输出 180 度相移时钟，CK270 为输出 270 度相移时钟，CK2X 为 2 倍频输出时钟，CK2X180 为 2 倍频相移 180° 输出时钟，DIV 为分频输出时钟。

<1>当 DLL 输入时钟为 10ns 时，DIV=1.5，DUTY\_CORRECT=TURE，输出波形为：

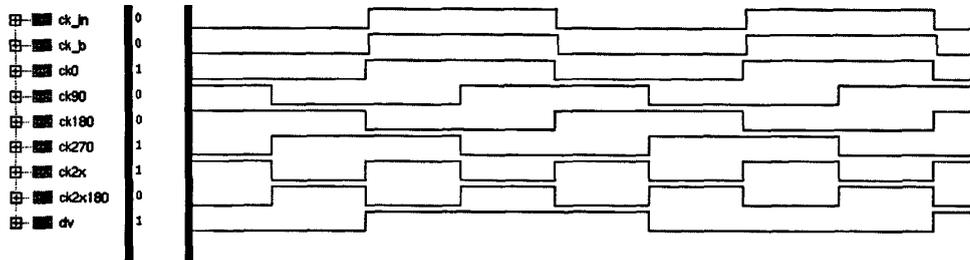


图 4-10 输入时钟为 10ns，DIV 配置成 1.5 的输出波形图

<2>当 DLL 输入时钟为 10ns 时，DIV=2，DUTY\_CORRECT=TURE，输出波形为：

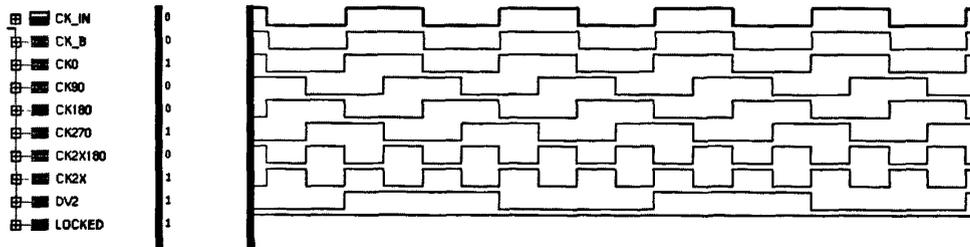


图 4-11 输入时钟为 10ns，DIV 配置成 2 的输出波形图

<3>当 DLL 输入时钟为 10ns 时，DIV=2.5，DUTY\_CORRECT=TURE，输出波形为：

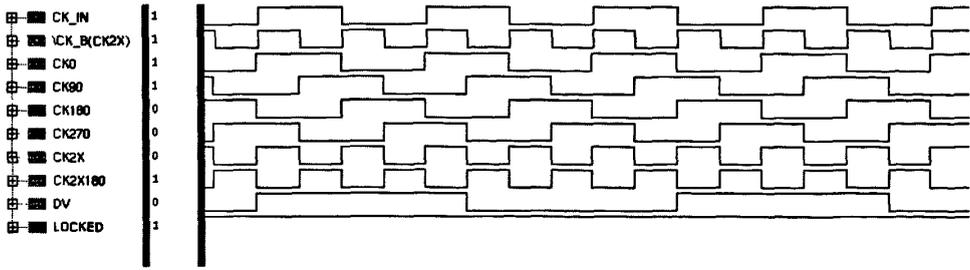


图 4-12 输入时钟为 12ns，DIV 配置成 2.5 的输出波形图

<4>当 DLL 输入时钟为 10ns 时，DIV=3，DUTY\_CORRECT=TURE，输出波形为：

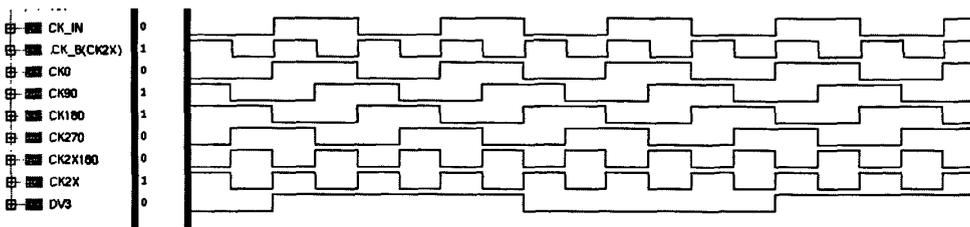


图 4-13 输入时钟为 10ns，DIV 配置成 3 的输出波形图

<5>当 DLL 输入时钟为 10ns 时，DIV=4，DUTY\_CORRECT=TURE，输出波形为：

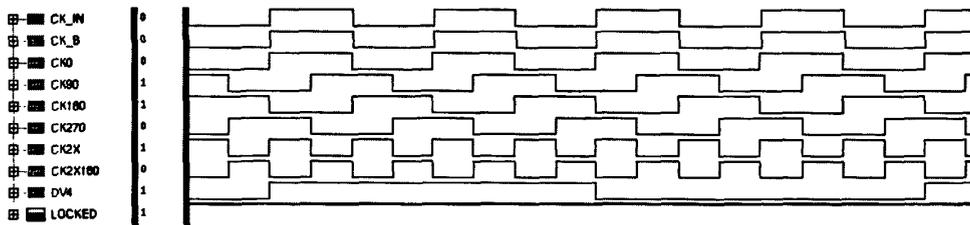


图 4-14 输入时钟为 10ns，DIV 配置成 4 的输出波形图

<6>当 DLL 输入时钟为 10ns 时，DIV=5，DUTY\_CORRECT=TURE，输出波形为：

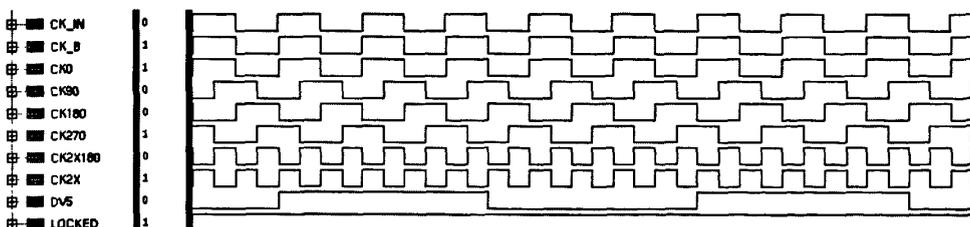


图 4-15 输入时钟为 10ns，DIV 配置成 5 的输出波形图

<7>当 DLL 输入时钟为 10ns 时，DIV=8，DUTY\_CORRECT=TURE，输出波形为：

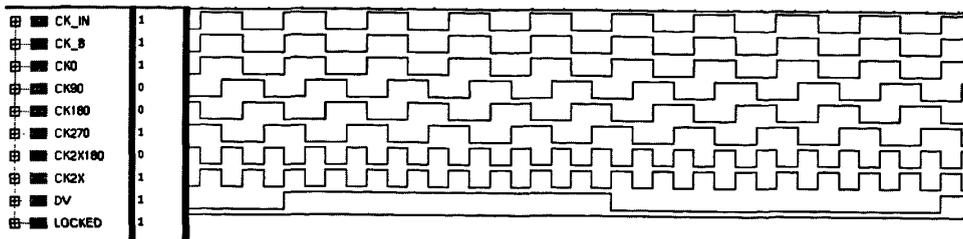


图 4-16 输入时钟为 10ns，DIV 配置成 8 的输出波形图

<8>当 DLL 输入时钟为 10ns 时，DIV=16，DUTY\_CORRECT=TURE，输出波形为：

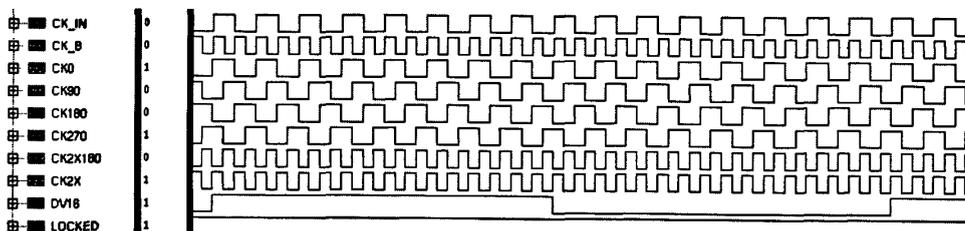


图 4-17 输入时钟为 10ns，DIV 配置成 16 的输出波形图

<9>当 DLL 输入时钟为 10ns 时，配置为 HFDLL 模式，DIV=16，DUTY\_CORRECT=TURE，输出波形为：

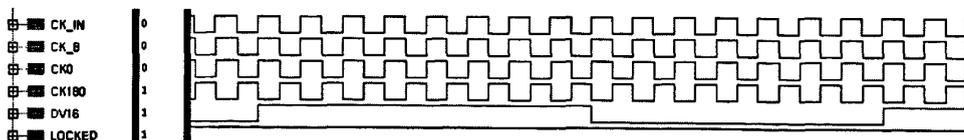


图 4-18 输入时钟为 10ns，HFDLL 模式下 DIV 配置成 1.5 的输出波形图

<10>当 DLL 输入时钟为 5ns 秒时，采用 HFDLL 模式，DIV=4，DUTY\_CORRECT=TRUE，输出各波形为：

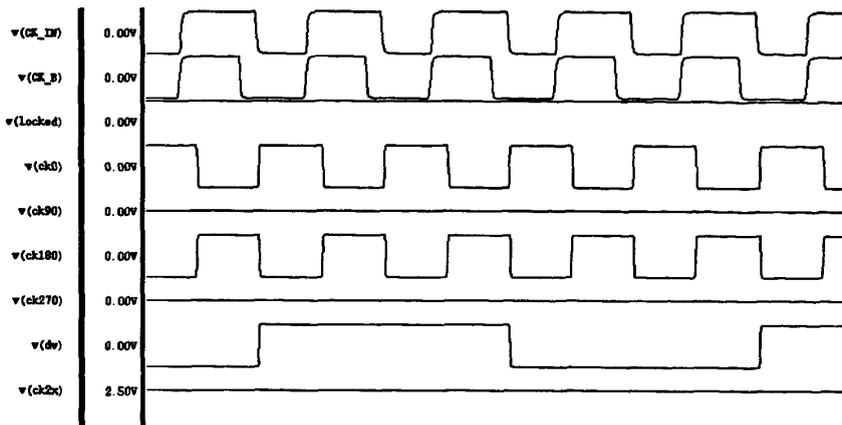


图 4-19 输入时钟为 12ns，HFDLL 模式下 DIV 配置成 4 的输出波形图

当输入时钟占空比不为 1:1 时，把输出波形配置成占空比为 1:1 的情况，查看输出波形是否满足

<11>当 DLL 输入时钟 CK\_IN 为 10ns(占空比为 3:7), DUTY\_CORRECT=TURE, 输出波形为:

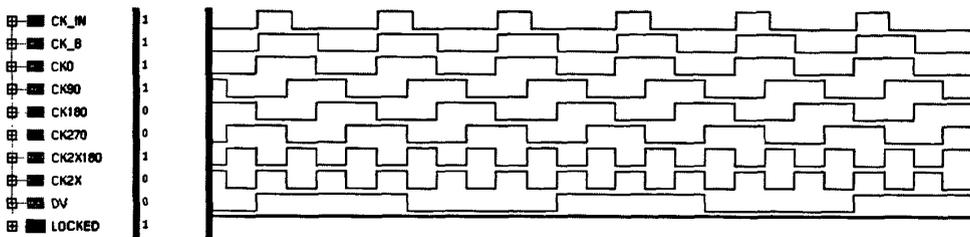


图 4-20 输入时钟占空比为 3:7，普通 DLL 模式下 DIV 配置成 4 的输出波形图

通过仔细的仿真验证，DLL 的功能都实现，它可以实现在 20MHz 到 200MHz 输入时钟的锁定与调整。其中在高频模式和普通模式下都很好的实现了对时钟的锁定与输出。

#### 4.5 混合仿真

混合仿真实际上是将模拟的 cdl 网表打包成 .v 的 verilog 网表，用来同 VCS 中的综合代码交互。在仿真过程中 cdl 网表进行 Hspice 仿真，VCS 中的代码加载 .sdf 文件后进行数字仿真，两者之间的接口信号，在经过转换后交互。数模混合仿真流程，如图 4-21。

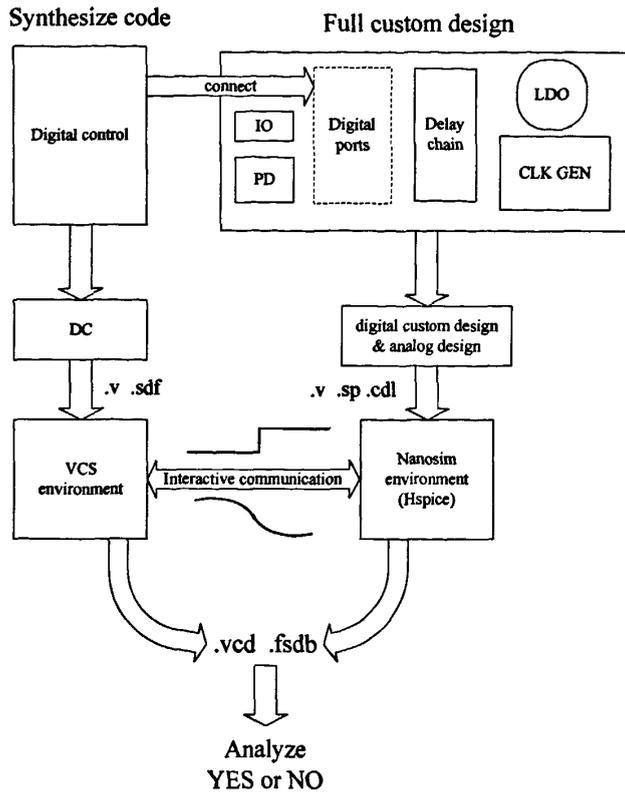


图 4- 21 数模混合仿真流程图

DLL 的数字控制部分是通过半定制设计流程设计完成的，鉴相器，延时链和时钟生成模块是通过全定制流程设计完成的。两者使用相同工艺，经过分别的设计仿真验证后，将设计数据（电路晶体管参数，版图参数）对接在一起做混合仿真。

后仿是通过 nanosim 和 vcs 的混合仿真实现的。

布图前后仿：将数字部分的代码，经过 DC 综合后，得到具有延时信息(.sdf)的.v 代码。这一阶段的数字部分模型已经具有器件延时，但连线的线载模型不包含延时信息。将全定制完成的其他模块（PD,DELAY\_chain, clk\_generator 等）通过.cdl 的网表形式导出，通过对网表进行.v 打包得到一个全定制模块的.v 文件。利用 nanosim 和 vcs 的混合仿真环境，将全定制模块的.v 文件指定在 nanosim 中进行 SPICE 仿真，将带延时参数的半定制数字模块.v 指定到 vcs 中进行数字仿真，两者之间的接口相互通信。

布图后仿真过程同布图前仿真相同，只是将版图的延时参数文件反标，器件和路径延时更加准确。

半定制数字模块的综合库： smic0.18

全定制 SPICE 仿真库： sm100001-1f.hspice

LDO 仿真库： spectre: sm100001-1f.scs

LDO 作为需要高精度仿真的模拟模块电路，通过 spectre 单独仿真。混合仿真时，LDO 的输出设定为恒定 2V，这样能都大幅度提高仿真效率。

#### 4.6 OSDLL 仿真结果

高频，低频两种模式下， OSDLL 和普通 DLL 的对比：

如图 4-22 为 OSDLL 和 DLL 后仿真锁定时间比较。两种模式下的 DLL 工作情况。输入时钟频率为 100M Hz。时钟两个锁定信号，时间差近 1us。在更低频的输入时钟情况下，OSDLL 的锁定速度比传统 DLL 要快几 u 至几十 u。在较快锁定速率的同时，又保证了调整机制的相同，以及调整准确性和时钟抗抖动等优势。

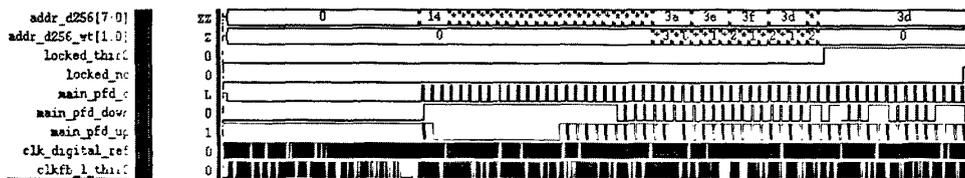


图 4-22 OSDLL 与传统 DLL 锁定时间比较

OSDLL 的特点是复用传统 DLL 的延时链，one-shot 译码逻辑相对简单，不过多的增加硬件开销，同时保持原 DLL 架构的优点，在多频段都能够提高锁定速度，频率适应性强。与传统 DLL 的锁定时间比较见图 4-23。图 4-23 中纵坐标代表锁定时间，横坐标代表仿真频率，百分数表示锁定时间相差的比例，可见在各个频率段，OSDLL 都能够明显的降低锁定时间。

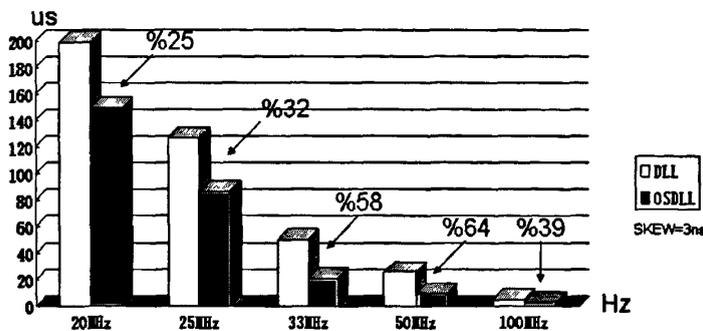


图 4-23 传统 DLL 与 OSDLL 各频段锁定时间对比

基于 SMIC 0.25um 工艺，设计 OSDLL 测试芯片。OSDLL 的工作频率在 20MHz 到 200MHz 之间，工作电压为 2.5 V。图 4-24 为版图。

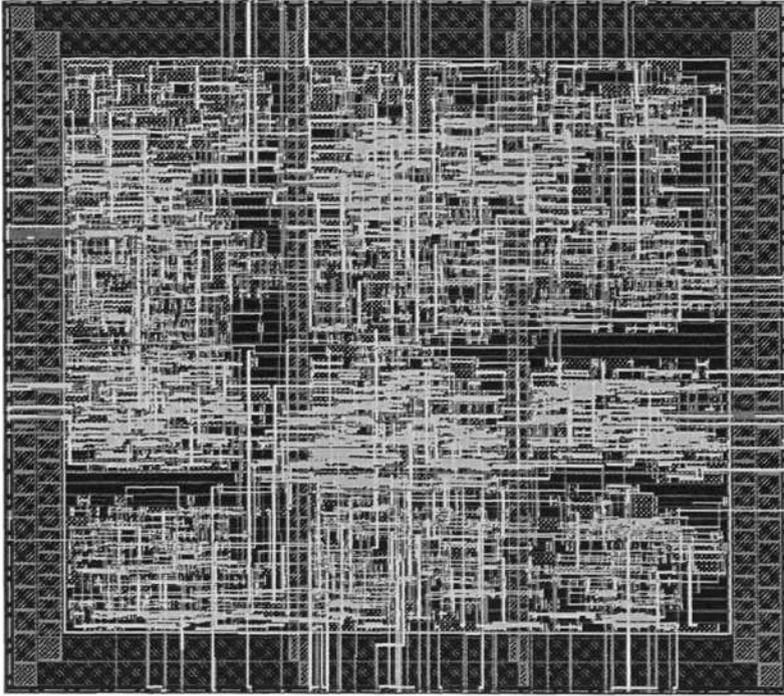


图 4-24 OSDLL 版图

#### 4.7 FPGA 芯片流片结果

基于 0.25um 5 层金属 CMOS 工艺，设计完成的 DLL 模块嵌入在 100 万门 SRAM 型 FPGA 芯片内部，作为其可选的时钟管理模块。如图 4-25，100 万门 FPGA 芯片测试 MPW 样片。此版本 FPGA 芯片采用经典的可编程 LUT（查找表），BRAM（块状 RAM），IOB(IO 模块），MUX（可编程路由），配置电路以及 DLL 模块组成，可编程资源达到 100 万门。具体可用逻辑规模，见表 4-5。

表 4-5 FPGA 芯片可用逻辑资源

Device	System Gates	CLB Array	Logic Cells	Maximum Available I/O	Block RAM Bits
100 万门 FPGA	108,904	20x30	2,700	180	40,960

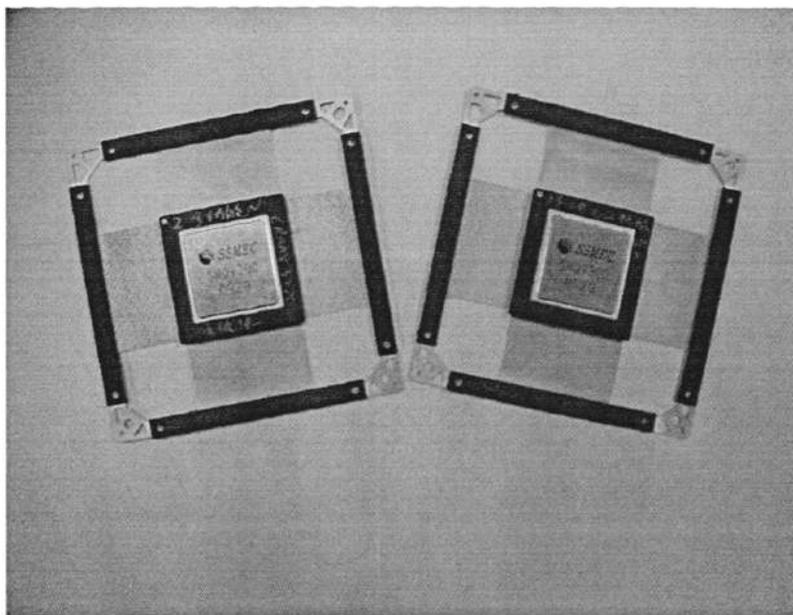


图 4-25 测试样片照片

#### 4.8 本章总结

本章介绍了 DLL 设计过程中的参数仿真设计，功能仿真，及混合仿真等。分别具体就 DLL 的整体架构参数仿真和功能仿真，PD 参数仿真，LDO 设计与仿真以及数模混合仿真平台等方面做了介绍。并展示了 FPGA 芯片流片结果。

## 第五章 DLL 模块多种环境下的应用

### 5.1 DLL 在 FPGA 中时钟管理功能

大规模 FPGA 芯片具有布线资源复杂，时钟树设计难度大的特点，当需要实现的用户设计电路在较高的运行速度下工作时，就必须具有合理的时钟分布网络。

在此版本 FPGA 中，我们设计采用经典的 H 树型时钟网络分布架构<sup>[27]</sup>与 DLL 控制调节的方式相结合。DLL 位于芯片版图位置的四个角，时钟输入 PIN 脚位于芯片的上端中央位置，图 5-1 是时钟网络的一部分示意图。当输入时钟频率较高时可以直接把输入时钟送入 DLL 的时钟输入端，把 DLL 的时钟输出连接到全局时钟输入 BUF，驱动整个时钟网络，然后把次级时钟分布网络的时钟接入 DLL 的时钟反馈端。因为二级时钟网络直接连接各单元模块，所以 DLL 可以通过调整使输入时钟 CK\_IN 和反馈时钟 CK\_FB 同步，即输入时钟沿与到达各单元模块的时钟沿同步，消除了由于驱动 BUF 延时以及负载众多而引起的时钟延迟、抖动和歪斜。同时，可以把 DLL 输出的分频、倍频或相移时钟通过网络的选择开关加载到不同的时钟网络上。

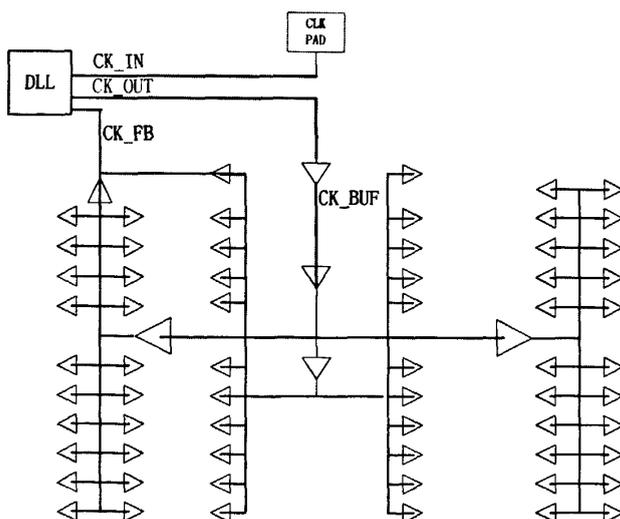


图 5-1 DLL 和 FPGA 时钟网络架构

除了芯片内时钟调节外，DLL 同样可以生成板级时钟信号源，为板级应用提供服务，如图 5-2。FPGA 在板级应用时，DLL 可以通过 FPGA 的 PIN 脚产生稳定时钟信号作为板级时钟源。

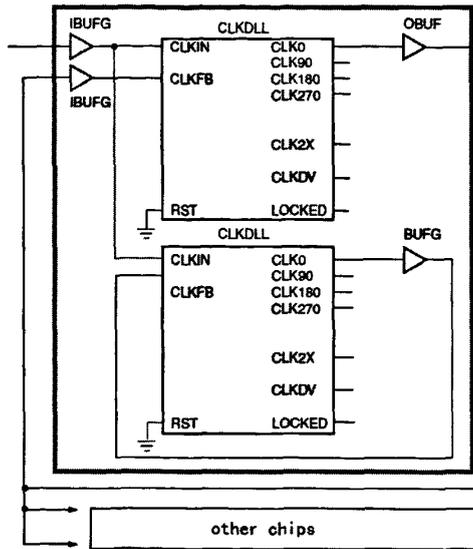


图 5-2 DLL 板级时钟信号调节

### 5.2 DLL 测试功能

RO 模式<sup>[28]</sup>, 即 Ring Oscillator 功能模式。当 DLL 处于 RO 配置模式下时, 对应配置选项, 可以选择 CLK90,CLK180,CLK270,CLK360 作为返回时钟, 构成长度不同的振荡器模式, 如图 5-3。

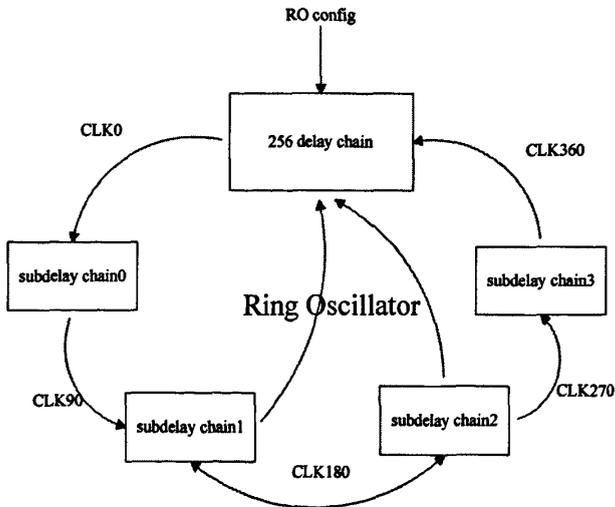


图 5-3 DLL Ring Oscillator (环震) 模式

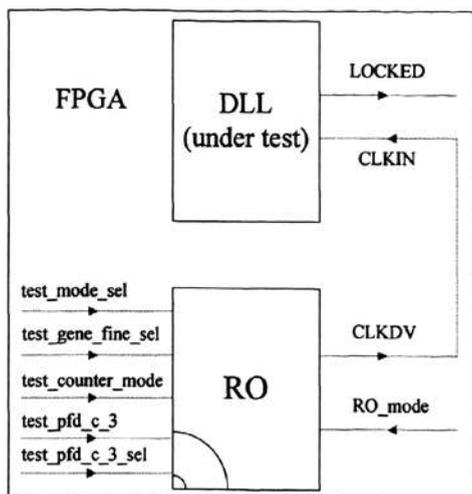


图 5-4 RO 测试模式系统

RO 测试模式可用来测试片内 DLL 的最低锁定频率，如图 5-4。FPGA 片内一个 DLL 模块做正常配置，作为激励源的 DLL（下）配置成测试模式，并且 RO 使能信号有效。测试模式下，可以通过外部测试控制信号，将内部计数器设置成任意计数值，通过计数器的不同计数值，调节延时链的延时单元数，从而改变环路振荡器的环形反相器数目，即改变振荡周期，实现不同的震荡频率。从某一个频率开始，通过 RO 的 CLKDV 输出激励时钟信号，作为待测 DLL 的时钟输入信号。

RO 测试模式系统选取 FPGA 内部一个 DLL 作为 RO，生成测试时钟源 CLKDV，作为待测 DLL 的输入时钟 CLKIN。在规定的时间内监测待测 DLL 的锁定信号 LOCKED。RO 不断的降低时钟输出频率，可以检测出待测 DLL 的最低锁定频率。

RO 模式测试仿真结果如图 5-5，图 5-6。测试模式下设定计数器的计数值，后停止输入时钟（kill=1），在 RO 模式下，DLL 开始自震荡，CLKDV 输出振荡器设定好的时钟信号。

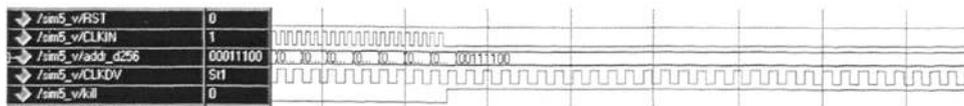


图 5-5 RO 功能仿真波形

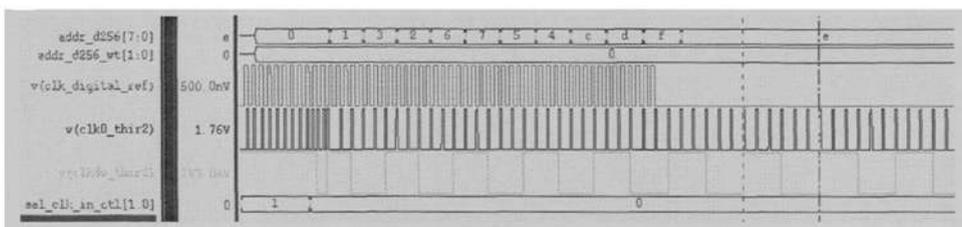


图 5-6 RO 后仿波形

### 5.3 FPGA 芯片在太空中的应用

随着我国航空航天事业的发展，FPGA 作为一种高效的设计应用芯片，在运载火箭，卫星和宇宙飞船的电路设计方面都有广泛的应用。然而太空辐照效应，尤其是 SEU（单粒子翻转）效应对 SRAM 型 FPGA 在太空中的正常工作有着极大的影响。如何抵抗 SEU 效应，成为 FPGA 设计在太空应用中的关键。

#### 5.3.1 SEU 效应

在 高能离子和质子的轰击下，SRAM 的存储内容极易发生变化，比如从逻辑“0”跳变到“1”，从而引起逻辑错误和功能失效，这就是所谓的单粒子翻转（SEU）效应<sup>[29]</sup>（图 5.7）。在太空的恶劣应用环境中，SEU 更加容易发生，严重威胁航空器材的正常工作。

SEU 主要对 FPGA 配置存储信息有较大影响，配置信息的改变进而会影响用户设计，对航空航天电子设备的稳定性有极大威胁。

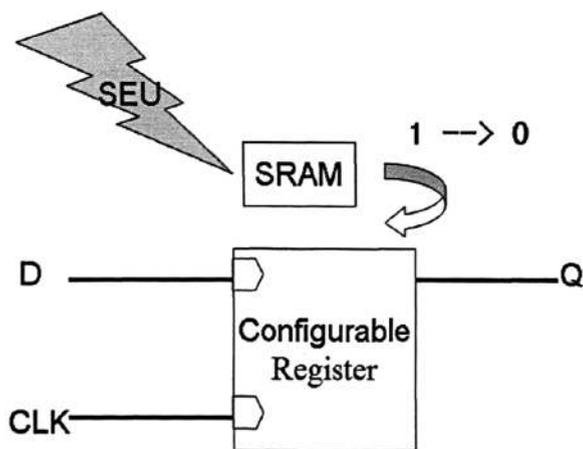


图 5-7 FPGA 内 SEU 效应

#### 5.3.2 TMR 技术

目前 FPGA 主要的抗辐照技术是 TMR(triple modular redundancy, 三模冗余

技术) [30]。所谓 TMR 技术即通过将目标电路拷贝三次并增加一个“投票表决”逻辑来加固敏感电路，如图 5-8。

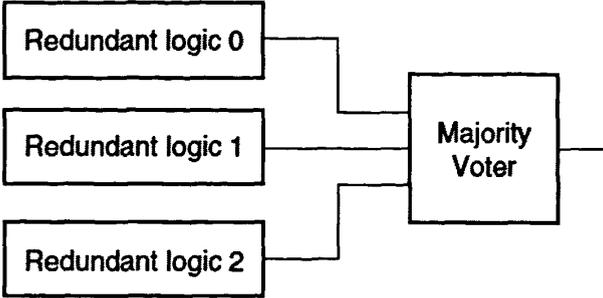


图 5-8 TMR 示意图

冗余逻辑[31]可能是单个寄存器或者整个逻辑设计电路。如果投票表决逻辑是多数表决器则当至少有两个冗余逻辑正常工作的情况下，能够正确输出结果。如果多数表决器的三个输入分别是 A, B, C, 输出是 V, 则  $V=AB+AC+BC$ 。

5.3.3 冗余技术在 FPGA 整体设计中的应用

FPGA 的 SEU 加固技术，需要在 FPGA 的整体设计中应用，才能取得理想的 SEU 缓解效果，利用冗余技术，结合选择性重配和周期性擦洗技术，在 FPGA 的设计中，我们依照的是如图 5-9 的设计构架图

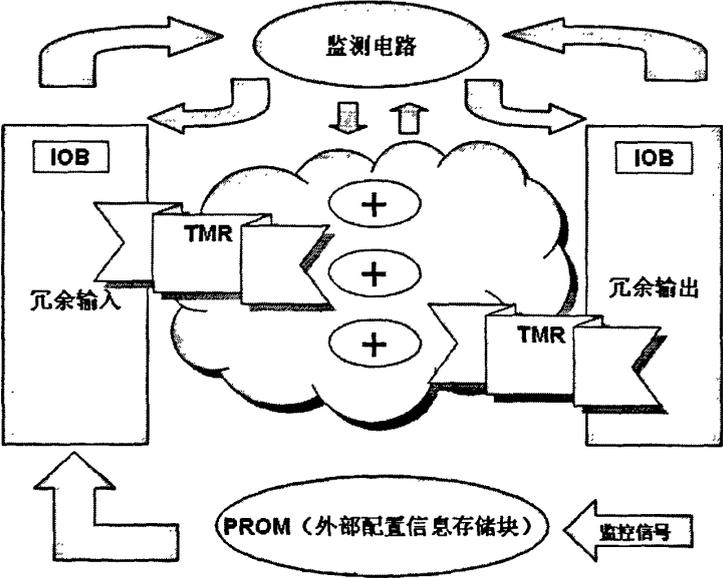


图 5-9 FPGA 的 TMR 加固技术

图 5-9 中，两端分别是输入/输出可编程 IOB 模块，中间的云图是 FPGA 设计资源，主要来说是 CLB 逻辑资源。监测电路和 PROM 分别利用被测 FPGA

之外的器件得到。通过对 FPGA 输入，输出，用户逻辑等多方面的冗余加固来确保系统在太空辐照中能够稳定工作，抵抗 SEU 效应。应当注意到太空应用中 FPGA 的稳定工作权重重要于硬件消耗，所以 TMR 技术尽管硬件资源占用率大仍广泛的应用于抗辐照设计中。下文将结合 FPGA 时钟管理电路 DLL 的 TMR 设计做进一步阐述。

### 5.3.4 DLL TMR 设计

针对我们设计完成的 FPGA 芯片结构，包括片内四个 DLL 和四个全局缓冲器 BUFGP。

通过 DLL 来连接 BUFGP 可以重新同步时钟信号，消除各自路径上的偏斜或者外部干扰造成的额外延时。然而 DLL 电路中的 SEU 会导致 DLL 失去同步调整的能力，例如使得 DLL 原本的配置失效，丧失原始调整目的，将可能会因此产生输出时钟抖动或者没有输出信号。

尽管 DLL 有 LOCKED 信号表征 DLL 已经完成同步调整，但这个信号不能用来作为 SEU 检测信号。因为当 DLL 发生单粒子翻转的时候必须经过复位（reset）才能重新完成同步调整。SEU 检测可以由图 5-10 中电路结构完成。

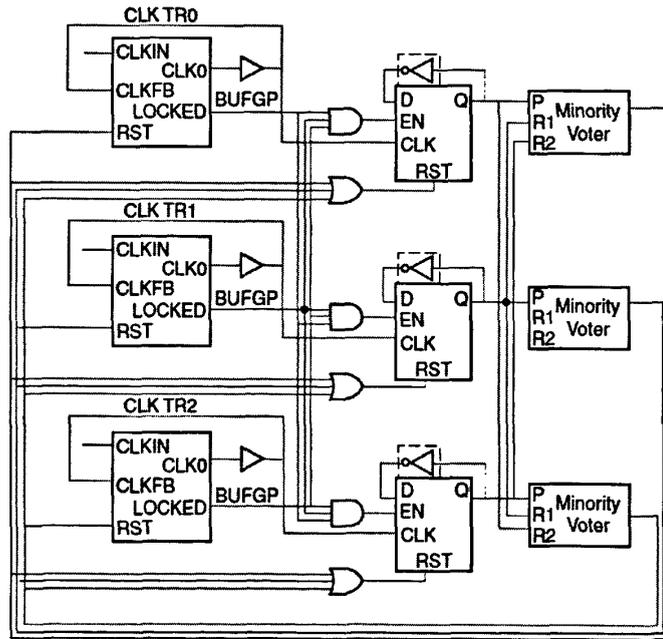


图 5-10 DLL 的 TMR 加固技术

当使用 DLL 作为时钟管理时，可以通过如下结构完成 DLL 的 TMR 设计以加固系统稳定。CLKIN 为系统时钟，输入到三个 DLL 的时钟输入端。DLL 经过三模冗余处理，及使用了 FPGA 片内三个 DLL 资源，DLL 的输出 CLK0 作

为反馈时钟经过全局缓冲器 BUFGP 反馈回 CLKFB, 三个 LOCKED 输入到“投票表决”逻辑中。

三个 1-bit 计数器用来监测三个时钟是否正常输出且同步。每个计数器分别由各自的冗余时钟做时钟驱动。当所有 DLL 都完成同步调整并且三个 LOCKED=1 时寄存器使能信号 ENABLE 信号有效。当三个寄存器的输出有一个不满足“多数投票”要求, “少数表决”逻辑将复位 DLL 以及三个寄存器。三个寄存器将不会在各自的时钟节拍下动作直至所有 DLL 达到锁存状态。

#### 5.4 本章总结

本章介绍了 DLL 的应用方面, 包括芯片级和板级的时钟管理应用, 以及根据 DLL 的可配置特点, 利用延时链将其配置成环震用来做片内测试。同时介绍了 FPGA 在太空中的抗辐照设计, 特别是 DLL 的 TMR 抗辐照加固技术。

## 第六章 结论和展望

### 6.1 结论

论文涉足于现场可编程逻辑芯片设计，这一国内尚属起步的领域，以一款 100 万门的 SRAM 型 FPGA 芯片的研发为基础，主要着力于 FPGA 片内时钟管理系统，即延时锁相环（DLL）的研究和设计。在 SMIC 0.25um CMOS 工艺下设计完成 FPGA 芯片，并经过功能及性能测试，结果显示 FPGA 功能正确，性能达到标准，特别是时钟管理模块 DLL 各项设计指标达到芯片使用要求。这是国内首款 FPGA 芯片流片成功，对于国内可编程器件的研发具有重要意义，填补了国内 FPGA 芯片设计和制造的空白。

论文从 DLL 的设计角度阐述了 FPGA 可编程器件的时钟管理技术原理及其设计应用等方面。总结论文所取得的成果如下：

1. FPGA 片内全数字 DLL 电路架构设计。论文从 DLL 原理阐述，电路结构设计，各子模块设计，仿真及验证方面介绍了 DLL 完整架构设计过程。其中数字控制逻辑依据半定制数字电路设计流程设计完成，鉴相器，可调延时链和时钟生成模块依据全定制流程设计完成。经过数模混合仿真平台，设计仿真通过，整合至 FPGA 芯片内部，在 SMIC 0.25um 工艺下，流片完成一款 100 万门 FPGA 芯片。经过测试 DLL 功能正确，在高低温等极限条件下能够正常工作。

2. 快速锁定 OSDLL 架构的提出和设计。论文在 FPGA 片内 DLL 的架构基础上提出了具有更快锁定速度的 OSDLL 架构，弥补了 DLL 锁定时间相对较长的缺点。结合 one-shot 技术，同时复用 DLL 的主延时链作为 one-shot 计算延时链，在不过多增加芯片面积的情况下较大程度的减少了 DLL 锁定时间，进一步提高了 DLL 和用户设计的系统性能。

3. DLL 设计的多领域应用考虑。可以将 DLL 模块作为一个 IP 嵌入 SOC 中，作为片上系统的时钟管理模块。在 FPGA 设计中可以优化用户设计时序，同时可以在板级应用中为其他芯片提供多种时钟源（移相，倍频，分频时钟等）。在航空航天领域，FPGA 芯片应用广泛，文中结合 TMR 技术介绍了 SRAM 型 FPGA 抗 SEU(单粒子翻转)等太空辐照效应的设计方法，主要论述了 FPGA 片内 DLL 的 TMR 设计。

目前芯片内的时钟管理模块仍然以 PLL 占据主导。PLL 具有时钟综合能力强，时钟锁定时间短，时钟调整范围大，可实现高频时钟同步和综合等操作的特点。但与此同时传统的 PLL 所带来的功耗较高，模拟电路设计占据消耗较大的芯片面积和对噪声敏感稳定性差，以及跨平台工艺可移植性差等缺点，都使得 PLL 在大型数字 FPGA 设计中难以使用。相对于 PLL，论文设计的 FPGA 片内 DLL 电路具有 20-200MHz 的调整时钟频率范围， $90^\circ$   $180^\circ$   $270^\circ$  相移时钟，倍频时钟，1.5-16 的整数/小数分频时钟输出的时钟综合能力，以及数字电路设计具有的功耗较低和较强的工艺移植性等优点。该 DLL 的高性能和可嵌入性，

大大的提高并且简化了高性能时钟的系统级设计，使 FPGA 在数字通信、计算机、自动控制等领域具有十分广泛的用途。

## 6.2 展望

目前国内的现场可编程逻辑器件芯片领域仍然处于起步阶段，此款 FPGA 芯片填补了国内的空白，其中 DLL 模块集成于 FPGA 芯片内部，配合 FPGA 时钟架构为用户提供强大的时钟管理功能。100 万门 SRAM 型 FPGA 已经流片，测试芯片结果显示 FPGA 芯片整体功能正确，主要性能达到设计要求。同国外主流 FPGA 厂商的芯片性能比较，此系列 FPGA 还有很长的路要走，但可以看到差距正在缩小，为后续更高性能 FPGA 芯片产品设计做准备，需要在以下几个方面进行改进研究和突破：

1.系统结构的改进：FPGA 系统结构的改进可以很大程度地改进系统的性能。比如优化时钟树结构，改进 IO 模块及 LUT 结构等。结构的优化能够使得路径延时减小从而帮助 DLL 改进时钟同步调整过程。

2.电路结构的优化：改进的电路结构适应于更小特征尺寸的工艺，便于进行工艺移植。在低功耗设计上作出更多的考虑，器件的动态功耗是电路设计考虑的主要因素之一，所以应该在保证电路工作性能的基础上，简化电路，减少电路规模。

3.设计参数的提高：高速时钟信号的生成和处理是现代通讯、处理技术的关键，它限制了信号处理的速度。所以生成高速、稳定的时钟信号是研究的主要目标，进一步的提高频率，降低时钟抖动性、歪斜，使其达到 GHz 水平。达到这个设计目标需要在时钟管理模块 DLL 和 FPGA 芯片整体性能上提升设计标准。

## 参考文献

- [1] Roland, E. Best. Phase-Locked Loops design, simulation and applications[M]. 北京: 清华大学出版社, 2003. pp.120-135
- [2] Remco C. H., Cicero S. Vaucher and Bram Nauta. Low-Jitter Clock Multiplication: A Comparison Between PLLs and DLLs[J]. IEEE transaction on circuits and system—II: analog and digital signal processing.2002, Vol. 49, NO. 8. pp.555 – 566
- [3] 胡华春、石玉编著.《数字锁相环路原理与应用》[M].上海: 上海科学技术出版社, 1990. pp250-271
- [4] Bruno W. Garlepp, Kevin S. Donnelly and etc.. A Portable Digital DLL for High-Speed CMOS Interface Circuits[J]. IEEE.Journal of solid-state circuits. 1999, Vol. 34, NO. 5. pp.632 – 644
- [5] Xilinx Inc.. Xilinx Programmable Logic Data book. www.xilinx.com, 2002
- [6] Altera Inc.. Altera Programmable Logic Data book. www.Altera.com, 2002
- [7] Underwood, Hemmert. Closing the gap: CPU and FPGA trends in sustainable floating-point BLAS performance[J]. FCCM 12th Annual IEEE Symposium on20-23 April 2004, pp.219 – 228
- [8] Silva, Ferreira. Exploiting dynamic reconfiguration of platform FPGAs: implementation issues[J]. IPDPS. 2006, 20th International 25-29, pp.8
- [9] Knack, Panel. Design Automation Tools for FPGA Design[J]. Design Automation, 1994, 31st Conference on 6-10. pp.676 – 676
- [10] David P. Chengson, Hansel A. Collins, Edward C. Priest, Scott W. Alvarez System and method to reduce jitter in digital delay-locked loops[P] U.S.: 5790612, Aug 4, 1998
- [11] XILINX Inc. Using the Virtex Delay-Locked Loop[J]. XILINX XAPP132. January 5, 2006. pp.2-3
- [12] Behzad Razavi. 《模拟集成电路设计》[M]. 陈贵灿等译.西安: 西安交通大学出版社, 2002. pp.432-437
- [13] 吴建辉著.《CMOS模拟集成电路分析与设计》[M]. 北京: 电子工业出版社, 2004. pp.302-314
- [14] Joseph H. Hassoun, F. Erich Goetting, John D. Logue. Delay\_lock\_loop\_with\_clock\_phase\_shifter[P]. U.S.: 6587534, Jul 1, 2003
- [15] Steven P. Young, John D. Logue. Digital phase shifter[P]. U.S.: 6775342, Aug 10, 2004
- [16] Andrew K. Percey. Digital spread spectrum circuitry[P]. U.S.: 7010014, Mar 7, 2006

- [17] Andy T. Nguyen. Clock doubler circuit and method[P]. U.S.: 6259283, Jul 10, 2001
- [18] Andy T. Nguyen. Counter-based duty cycle correction systems and methods[P]. U.S.: 6788120, Sep 7, 2004
- [19] Andy T. Nguyen. ONE-SHOT DLL CIRCUIT AND METHOD[P]. U.S.: 6255880, Jul 3, 2001
- [20] Andrew K. Percey, San Jose. Glitchless delay line using gray code multiplexer[P]. U.S.: 6400735,Jun.22,1998
- [21] Paul G. Hyland et al. Delay line circuit providing clock pulse width restoration in delay lock loops[P]. U.S.: 6788119, Sep 7, 2004
- [22] F.Erich Goetting Paul G. Hyland Joseph H. Hassoun. PRECISION TRIM CIRCUIT FOR DELAY LINE[P]. U.S.: 6204710,Jun.22,1998
- [23] Gabriel A. Rincon-Mora, Phillip E. Allen. Optimized Frequency-Shaping Circuit Topologies for LDO's[J]. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING, VOL. 45, NO. 6, JUNE 1998,pp.703-708
- [24] Chaitanya K. Chava, José Silva-Martínez. A Frequency Compensation Scheme for LDO Voltage Regulators[J]. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 51, NO. 6, JUNE 2004, pp.1041-1050
- [25] Andy T. Nguyen, Jack Siu Cheung Lo. Clock divider circuit with duty cycle correction and minimal additional delay[P]. U.S.: 6744289,Jun 1, 2004
- [26] Andy T. Nguyen. Fast-locking DLL circuit and method with phased output clock[P]. U.S.: 6501312, Dec 31, 2002
- [27] John P. Uyemura著.周润德译.《超大规模集成电路与系统导论》[M].北京: 电子工业出版社, 2005. pp.390-455
- [28] Siuki Chan. Measuring a minimum lock frequency for a delay locked loop[P]. U.S.: 6502050, Dec 31, 2002
- [29] Keith W. Golke, Paul S. Fechner. SEU hardening circuit[P]. U.S.: 6058041, May 2, 2000
- [30] C. L. Axness, J. R. Schwank, P. S. Winokur. SINGLE EVENT UPSET IN IRRADIATED 16k CMOS SRAMs[J]. IEEE Transactions on Nuclear Science, Vol. 35, No. 6, December 1988, pp.1602-1607
- [31] Keith S. Morgan, Daniel L. McMurtrey, Brian H. Pratt. A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs[J]. IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 54, NO. 6, DECEMBER 2007, pp.2065-2072

## 攻读硕士学位期间发表的论文

1. 王忠涛, 杨明武. 可实现快速锁定的 FPGA 片内延时锁相环设计. 电子科技, 2010 年第 23 卷第 4 期