

基于FPGA任意分频系统设计

The Design of a Frequency Division System Based on FPGA

摘要: 本文给出了分频技术通用模型。并结合最新的一些分频技术,提出了一种基于FPGA全新的分频系统的设计方法,简单的介绍了设计的思路、原理及其算法,并对该方案的性能进行了分析,给出了仿真波形,RTL综合电路图。

关键词: 同步预处理;FPGA;整数分频;小数分频

引言

随着无线电技术的迅猛发展,雷达、导航、宇宙飞船、导弹以及空间探索工作的开展,需要高精度、高稳定度的频率源。这些频率源不但造价昂贵,而且它们基本上都只能输出单一频率。而目前在生产实践和科学研究中,却往往要求获得大量的高稳定度的频率。随着超大规模可编程器件的出现,人们往往从一个或多个标准频率源,通过分频技术,得到自己所需要的频率,为此我们设计一个分频系统的通用模块,它能够进行任意分频,由于采用的分频技术是利用数字运算(加、减、乘、除)产生的,因此频率稳定,分频速度

快,可以满足大多生产和科研要求。

通用分频系统模型

通用分频系统由四大模块组成,第一部分是同步预处理;第二部分是技术器/累加器模块;第三部分是算法部分;第四部分是选择输出。其结构如图1所示。

同步预处理

首先,我们对输入分频时钟(in_clk)进行预同步,对输入信号进行同步复制,使得输入信号与系统时钟(SYSCLK)达到同步。在理想状态下,如果SYSCLK频率:Fsys=128MHz,则系统时钟周期(单位:ns)为:

$$T_{sys} = 109/128 \times 106 = 125/16ns$$

(1)

分频in_clk频率:Fin_clk=44KHz,则in_clk时钟周期(单位:ns)为:

$$T_{in_clk} = 109/44 \times 103 = (25/11) \times 104ns$$

那么通过计算,用系统时钟对in_clk的复制,在一个in_clk周期内,由于系统时钟精度所产生的系统标准误差esys范围区间是:

$$esys [-7.1023ns/T_{in_clk}, 0.7102ns/T_{in_clk}]$$

上面分析的是同步预处理的理论过程,下面重点介绍,同步预处理的实现过程。

在图1中波形中,首先可以看到系统进行同步预处理的过程,即提取分频信号的上升沿和下降沿标志信号,并让该上升沿和下降沿做到与系统时钟sysclk同步。

下面我来说明程序是怎样从clk_in中取出clk_upedge、clk_downedge。首先让clk_in通过一个时钟为sysclk的D触发器,这样得到clkreg1信号,在把

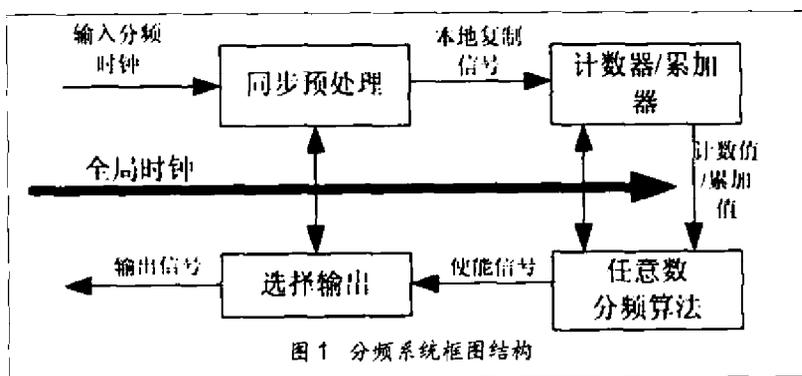


图1 分频系统框图结构



clkreg1通过一个时钟同样为sysclk的D触发器得到信号clkreg2,然后在取clkreg1的高电平和clkreg2的低电平的时候,取出一个clk_upedge的高电平,其它为低电平;同样可以在clkreg1的低电平和clkreg2高电平时,取出一个clk_downedge的高电平,其它为低电平。因为整个提取过程都是在系统时钟的上升沿进行的,所以提取得到上升沿和下降沿能够做到和系统时钟同步。从图2中可以看到clk_downedge、clk_upedge、clkreg1、clkreg2的仿真波形。实现电路图见图3。

累加器/计数器

通常在分频技术中经常使用的最

多的器件就是计数器/累加器,通过计数器的计数功能可以完成整数的任意分频;通过累加器的运算,可以完成小数的任意分频。在后我会详细介绍计数器/累加器在这两种分频技术运用。

算法部分

算法部分是整个通用分频器的核心,如整数分频50%的占空比,小数分频的实现,这些要求全部使用算法实现的,因此一个算法的好坏,就决定了程序分频器的直接性能指标。在后会详细描述两种分频技术的算法。整数比较简单,小数相对来说比较复杂。

选择输出

当完成了各类运算后,就需要把

得到的结果输出,选择输出在系统时钟的驱动下完成信号的输出。

任意整数分频技术

整数分频技术是整个分频技术中最基础的分频技术,其中整数分频包括偶数分频和奇数分频。下面介绍一种经典的任意整数分频技术,它可做到占空比50%的要求。首先,利用倍频技术,把clk_upedge、clk_downedge合并成一个信号du_clk,这样就可以得到一个频率是clk_in两倍的信号,在利用计数器在du_clk的时钟下计所需要整数个时钟的高电平,所需整数个时钟的低电平,就可以顺利地把整数分频分出来了,并且可以获得50%的占空比。仿真波形如图4所示。RTL电路图如图5所示。

整数分频性能分析

1. 延迟时间。因为是以系统时钟为参照,从RTL综合电路图可以看到整个系统,在sysclk的驱动下工作,因此系统是一个同步系统,所以本地复制信号与输入参考信号的相位误差最大为一个SYSCLK周期。我们使用的是Altera公司的stratix系列EP2S15F672C4芯片,系统时钟频率最高能达到217.44MHz,最大误差为4.599ns。

2. 整数分频范围。因为采用任意整数分频算法,分频后输出信号的频率范围可以从0Hz到系统时钟的最高频率的一半。对于分频的范围,可以做到一万以上都没有问题的。当然选择范围值还要根据芯片的逻辑单元多少来定。

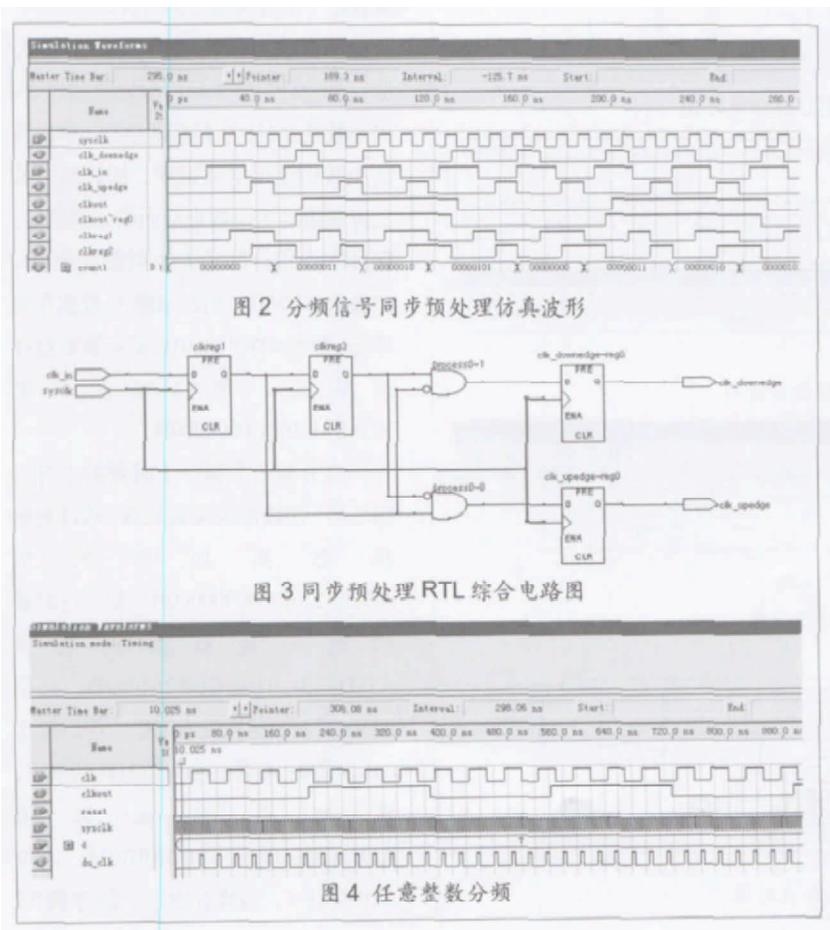


图2 分频信号同步预处理仿真波形

图3 同步预处理RTL综合电路图

图4 任意整数分频

任意小数分频

小数分频技术相对来说比较复杂，因为该分频技术存在一个算法问题。下面首先来介绍算法的理论过程。假设小数分频比是Q，分频器输入脉冲数是N，分频器输出脉冲数是M，则

$$Q = N / M$$

另外由于小数分频不要求占空比，因此只要在M个周期里面取出N个周期的时钟就可以了。另外

$$N = Q * M$$

所以可以设计一个寄存器里面放Q值，乘M，就相当于把Q值累加M次，这样就可以得到N的值了，从而可以实

现分频比的要求了。

下面详细的介绍算法的实现过程。该算法需要的器件，主要三部分组成：累加器、逻辑控制电路、脉冲输出电路。

累加器组要有寄存器和加法器构成，主要完成的任务是不断的累加运算。逻辑控制电路，主要的任务是完成累加器的高位溢出处理，所谓的高位溢出处理主要包括两个步骤：1 先判断高位是否溢出。2 然后处理高位的两种情况，如果溢出，则需要做两项处理：第一、输出一个分频信号的时钟；第二、把溢出位清零其它位保持不变；反之，这输出一个低电平，其余保持不变。输出电路完成的任务是当高位为溢出时，完成输出一个分频信号的时钟。

下面将举例说明该算法的流程。假设要设计一个3/4分频的分频器，3/4的小数是0.11，为了运算方便，本次算法中我们采用定点运算。因此输入的二进制值均为小数点后面的二进制数，例如本次设计的输入数据宽度是32，用数组NUM[31..0]表示输入数据存放单元，数组ADD[32..0]表示运算数据存放单元，所以输入数据NUM[31..0]=C0000000H。

当分频信号第一个时钟的上升沿到来后，把数据送入累加器，假设初始状态累加器的值ADD[32..0]=00000000H，那么这时累加器的运算值就变成ADD[32..0]=0C0000000H，高位ADD[32]=0，运算没有溢出，所以输出outclk为0；当第二个分频时钟到来后，累加器的运算值ADD[32..0]=180000000H，高位ADD[32]=1，运算有溢出，

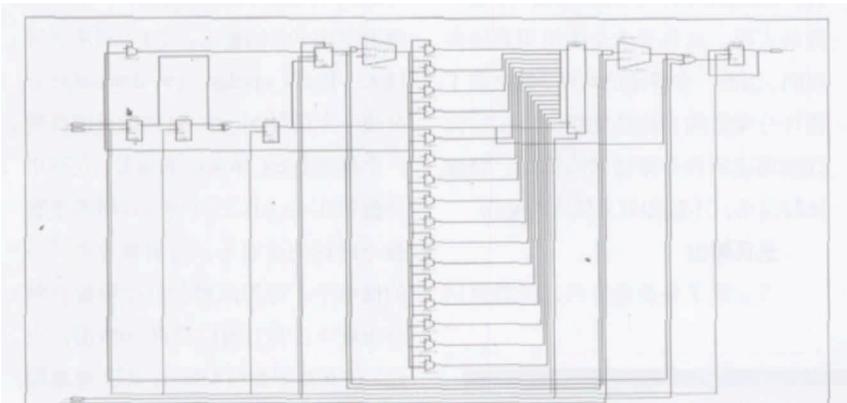


图5 整数分频RTL综合电路图

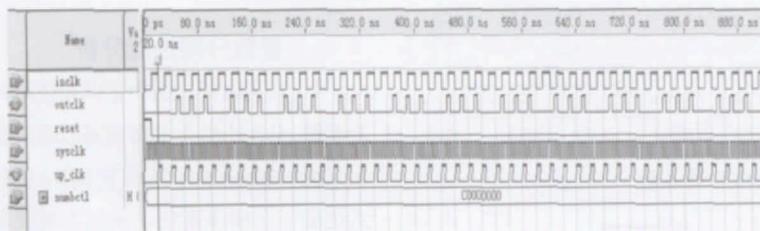


图6 小数分频仿真波形

Timing Analyzer Summary								
Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1 Worst-case tsu	N/A	None	5.368 ns	numbcd[3]	outclk~reg0	--	sysclk	0
2 Worst-case tco	N/A	None	5.977 ns	outclk~reg0	outclk	sysclk	--	0
3 Worst-case th	N/A	None	-2.105 ns	inclk	clk_delay1	--	sysclk	0
4 Clock Setup: 'sysclk'	N/A	None	312.99 MHz (period = 3.195 ns)	numbcd[0]	outclk~reg0	sysclk	sysclk	0
5 Total number of failed paths								0

图7 系统时序分析

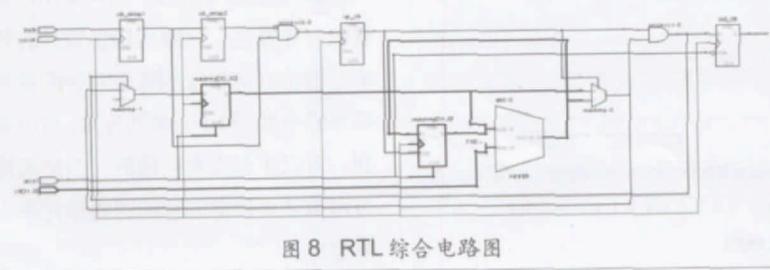


图8 RTL综合电路图

Field Applications

