

时钟及时钟域

转载yundanfengqing_nuc2020-05-08 15:35:531103收藏12版权

文章目录

前言

时钟及时钟域

- 时钟，时序逻辑的心跳
- 时钟信号基本特征
 - 时钟信号基本特征参数介绍
 - 如何区分时钟和数据
- 时钟信号的分类
 - 按来源分
 - 外部时钟
 - 再生时钟
 - 门控时钟
 - 行波时钟
 - 按波形分
 - 连续时钟
 - 间歇时钟
 - 不规则时钟
- 时钟域
 - 时钟域的概念
 - 时钟树简介
 - 时钟树的分类
 - 正确的时钟使用方式
 - 什么情况下时钟应该“上树”？
 - 如何选择时钟树？
 - 时钟信号怎么“上树”？
 - 使用全局时钟树资源
 - 方法一，通过正确的物理连接。
 - 方法二，通过恰当的代码描述。
 - 使用区域时钟树、IO时钟树资源
 - 被“拉下树”的时钟信号

前言

本文节选自《FPGA之道》，通过作者的讲解，一起来看看时钟和时钟域的相关问题，下一篇博客再看看作者如何来解释跨时钟域问题。

时钟及时钟域

时钟，时序逻辑的心跳

点赞1评论分享收藏12举报

关注一键三连

在时序逻辑中，正是时钟信号将各个存储单元中的数据一级、一级的推动下去，如果时钟信号突然停止，那么整个时序逻辑也将陷入瘫痪，因此，时钟就好像时序逻辑的心跳一样，那么重要、却又那么平常地存在着。

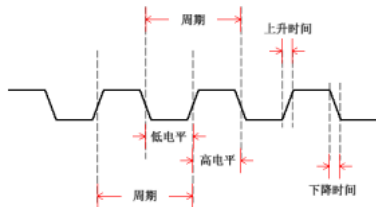
几乎所有的FPGA设计都是时序逻辑，因此几乎所有的FPGA设计都离不开时钟信号，但是正因为时钟信号是时序逻辑最基本的必须，就好比空气之于众生一样，也许有些时候，你并不能意识到它的重要性，而这种潜意识里对时钟信号的不重视，则早已为FPGA设计的最终失败埋下了深深地伏笔。因此，要想确保FPGA设计最终的成功，先要确保时钟信号的成功！如果有人让你评判在FPGA上实现某一个功能的可行性，第一步，看时钟，判断该功能对时钟信号的要求是否超越了FPGA的能力范围；如果让你着手开始一个FPGA设计，第一步，还是先看时钟，选取合适的时钟方案作为整个项目的支撑。

时钟的重要性远非区区一个章节所能介绍的完的，可以说，本书的大部分内容都在围绕着时钟信号展开，不过在本章节，我们将目光更加集中于时钟信号本身，来对其进行一些基本介绍，好让大家更加清楚的认识时钟。

时钟信号基本特征

时钟信号基本特征参数介绍

时钟信号具有什么样的基本特征呢？本小节将为大家慢慢道来。请看下图：



对于一个时钟信号来说，最重要的三个参数非周期、频率和占空比莫属了。

周期一般用大写的英文字符T表示，其的数学定义为：对于一个函数f(x)，如果存在一个非零常数T，使得当x取定义域内的每一个值时，都有 $f(x+T) = f(x)$ ，那么f(x)就叫做周期函数，而非零常数T则叫做这个函数的周期。由此可见，一个函数可以有很多个周期，通常，我们取其中最小的非零常数T，即最小周期做为该周期函数的周期。由此可见，连续两次上升沿或下降沿之间的时间间隔即为时钟信号的周期，如上图所示，当然了，时钟信号周期的认定并不局限于此。

频率一般用小写的英文字符f表示，它和周期互为倒数，即 $f = 1 / T$ 。例如，如果时钟信号的周期为0.1秒，那么其频率就为10赫兹，表示一秒钟的时间内共包含了时钟信号的10个周期。由此可见，相比于周期参数，频率参数在描述时钟信号变化的快慢程度上更加直观。

占空比等于时钟信号一个周期内逻辑1持续的时间与周期时间的比值。对于数字波形来说，波形图上几乎除了逻辑1就是逻辑0，而上图中的“高电平”、“低电平”即分别对应一个时钟周期内逻辑1和逻辑0的持续时间。通常来说，50%是最常见的时钟信号占空比，即“高电平”宽度等于“低电平”宽度。

除了周期、频率、占空比之外，上升时间和下降时间也是时钟信号中比较重要的特征参数。由于现实世界中，电容的充、放电等等都需要一定的时间程，因此不同逻辑电平之间的过度总是需要有一个过程，而上升时间就是指时钟信号从逻辑0变化到逻辑1所需要的时间，下降时间则是指时钟信号从逻辑1变化到逻辑0所需要的时间，分别如上图所示。对于理想的时钟信号，我们说时钟的上升沿就是从逻辑0跳变到逻辑1的这个瞬间，时钟的下降沿则是从逻辑1跳变到逻辑0的这个瞬间。而对于实际的时钟信号来说，由于电平之间的变化需要消耗时间，因此，上升沿和下降沿发生的具体时刻就不太好确定，而要根据接收时钟信号的数字器件对于逻辑电平的判决门限来推算了，不过可以肯定的是上升沿、下降沿一定位于上升时间、下降时间之内。对于时序逻辑来说，其内部的绝大部分存储单元都是边沿敏感的，例如寄存器、BLOCK RAM等等，因此一定要对时钟信号的上升沿、下降沿有一个清楚的认识。

如何区分时钟和数据

在数字的世界中，所有的信号都是在逻辑0与逻辑1之间不断切换的，而不仅仅限于时钟信号。因此，数据信号也完全可以具有时钟信号的所有基本特征，那么此时，我们该如何分辨该信号是时钟信号还是数据

点赞1

评论

分享

收藏12

举报

关注

一键三连

如果仅仅通过观察信号的数字波形，是无法分辨一个信号到底是时钟信号还是数据信号的。要想做出准确分辨，必须去查看该信号在数字电路中的连接关系，举个例子，如果一个信号连接到一个寄存器的时钟端，那么它就是一个时钟信号；如果该信号连接到一个寄存器的数据输入端，那么它就是一个数据信号；如果一个信号连接到一个寄存器的时钟端的同时又连接到另一个寄存器的数据输入端，那么它也就具有了时钟信号与数据信号的双重身份。

由此可见，时钟信号的本质，是在于其是否为时序逻辑提供“心跳”机制，而不在于其具体的表现形式。

时钟信号的分类

按来源分

时钟信号按照其来源分，可分为外部时钟和内部时钟，而内部时钟又可分为再生时钟、门控时钟、行波时钟，分别介绍如下：

外部时钟

外部时钟，指时钟信号的来源是在FPGA芯片的外部。通常来说，外部时钟源对FPGA设计来说是必须的，因为一般FPGA芯片内部没有能够产生供内部逻辑使用的时钟信号的选择频和激励电路。所以通常来说，我们需要在FPGA芯片的外部使用晶振以及恰当的电阻、电容、电感、三极管等器件，来搭建用于产生时钟信号的电路，并将其通过FPGA芯片的恰当物理管脚引入到FPGA内部供时序逻辑使用。

顺便说一句，当我们用示波器观察外部时钟信号时，可以发现其波形很可能不是一个规则的方波，而是一个正弦波。这是由于方波信号实际上是一个频带非常宽的信号，因为其有着尖锐的上升沿与下降沿，而晶振的选择回路只能针对某一个频率的信号进行放大，而对其他频率的信号都会衰减，再加上任何信道其实都相当于一个滤波器，所以导致时钟信号的带宽非常的有限，因此都类似表现为正弦波，这也是时钟信号中为什么会有“上升时间”和“下降时间”的一个原因。不过模拟的时钟信号经过逻辑门的驱动后，由于MOS电路的陡峭导通特性，会对时钟信号的“上升时间”和“下降时间”起到积极的缩减作用。

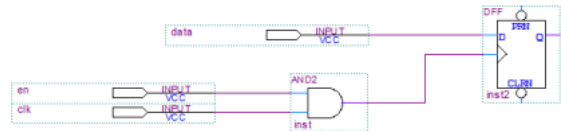
再生时钟

再生时钟，指FPGA内部产生的新时钟，当然了，这并不是凭空产生的时钟（因为FPGA芯片内部一般不具有产生时钟信号的电路结构），而是以一个输入的时钟信号作为参考，然后在此基础上通过调整其频率和相位而产生出来的新时钟。

目前来说，FPGA芯片内部能够产生再生时钟信号的模块主要有DCM和PLL两个，在【本篇->编程思路->DCM与PLL】章节中，我们将为大家详细介绍。顺便说一下，FPGA芯片内部仅有DCM和PLL可以对时钟信号进行升频操作。

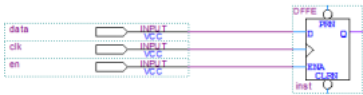
门控时钟

门控时钟，指的是由组合逻辑产生的时钟，其中，组合逻辑的输入可以全部是数据信号，也可以包含原始时钟信号。由于组合逻辑中的基本单元就是与、或、非等门电路，而与门和非门又具有“开关性”，固该类时钟又称为门控时钟。例如，下面的电路图门控时钟是由与门产生的，当en信号为1时，与门打开，原始时钟信号可以通过并作用到触发器；而当en信号为0时，与门关闭，触发器由于没有时钟信号驱动，将会处于休眠状态，不会有任何动作：



通常情况下，不建议大家在FPGA设计的内部引入门控时钟，因为门控时钟由组合逻辑产生，那么它身上就潜伏了组合逻辑的最大隐患——竞争与险象，所以门控时钟信号容易产生毛刺，而寄存器等存储单元对时钟信号的边沿都非常的敏感，因此具有毛刺的时钟会造成时序逻辑的不稳定。

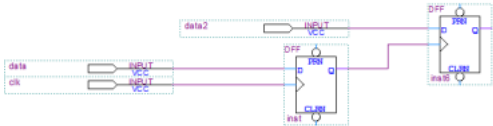
事实上，控制时钟信号的目的是为了控制触发器的行为，这点我们也可以通过控制触发器的使能端来做到，因此，上例中的电路完全可以修改如下：



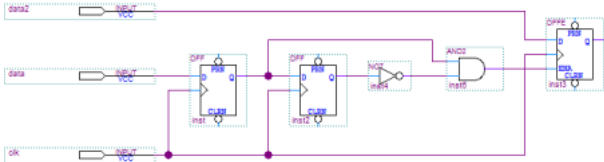
这样一来，这个系统将会免受竞争与险象的困扰。

行波时钟

行波时钟，指的是由时序逻辑产生的时钟。例如，A寄存器的输出如果作为B寄存器的时钟，则A寄存器的输出称为行波时钟。例如，如下电路中，驱动右边那个触发器的时钟即为行波时钟：



通常情况下，也不建议大家在FPGA设计的内部引入行波时钟，因为这样会在FPGA设计中引入新的时钟域（后续小节将会介绍），增加时序分析的难度，并且由于行波时钟的相位通常会滞后于原始时钟，因此后续触发器的保持时间（关于保持时间的概念参见【时序分析篇->基本概念介绍->常用时间参数介绍】小节）不一定能够得到满足。事实上，采用行波时钟的目的无非是为后续时序电路的处理速度进行降频，而要实现降频的功能，除了通过降低时钟信号的频率外，仍然可以通过控制后续时序电路存储单元的使能端来实现，因此，上例中的电路完全可以修改如下：



这样一来，整个时序逻辑将只被一个时钟信号所驱动，变得清晰又易控。

按波形分

时钟信号按波形分，可分为连续时钟、间歇时钟、不规则时钟，分别介绍如下：

连续时钟

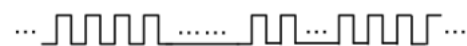
连续时钟，指从波形图上来看，时钟波形呈连续的、周期的，例如下图所示：



FPGA内部的时钟通常来说都是连续时钟。

间歇时钟

间歇时钟，指从波形图上来看，时钟波形存在间断，时有时无，例如下图所示：



间歇时钟常见于FPGA的数据输入或输出接口，这是因为有些外围器件的同步接口不包括使能端，因此当没有需要传输的数据时只能通过关闭时钟来暂停通信，并且这样也可以在一定程度上减少功耗。

不规则时钟

不规则时钟，指从波形图上来看，时钟的波形无固定规律可循，没有固定的周期、频率，也没有固定的占空比，并且还



不规则时钟也常见于FPGA的数据输入或输出接口，与间歇时钟产生的原因类似，只不过间歇式时钟往往针对成包或成帧传输数据的接口，因此每次都会输出一连串规则的波形信号，而不规则时钟针对的接口数据量往往比较小，也没有固定的数据结构，因此有一个数据就会传一个数据。

时钟域

时钟域的概念

时序逻辑离不开时钟，有时钟的地方也必然存在着时序逻辑，而随着FPGA芯片所承载功能的日渐复杂与多样化，单个时钟信号已经往往不能满足FPGA设计的需求。就拿一个最简单的串并转换接口来说吧，要实现它就至少需要两个时钟信号，因此，通常情况下，一个FPGA设计中往往要用到多个时钟信号。都说“一个和尚挑水喝，两个和尚抬水喝，三个和尚没水喝”，当FPGA设计中的时钟增多了之后，会不会也存在一些问题呢？答案是肯定的！多时钟并存，最严重的就是会导致跨时钟域问题，这个我们将会在下一小节详细介绍，而现在，让我们将目光集中于时钟域上，只有先搞清楚什么是时钟域，后续的讨论才会变得有意义。

简而言之，时钟域就是时钟信号的“势力范围”，用来昭告天下，“哪一片归我管”，“这里我最大”等等。一个时钟域中只能存在一个时钟信号，典型的“一山不容二虎”，但一个时钟信号最多可以对应两个时钟域，当其上升沿和下降沿分别都被一部分资源敏感的时候。而被时钟域所瓜分的资源，正是具有存储功能的各个单元，而其中最典型的的就是寄存器。要想判断一个寄存器是属于哪一个时钟域的，方法很简单，只要看它的时钟输入端口接的是哪个时钟信号以及敏感哪个边沿即可。时钟信号直接掌控着属于其时钟域内的寄存器们，同时也间接的掌控着一些组合逻辑资源，因为FPGA内部组合逻辑的输入往往是寄存器的输出。不过，要想判断某一块组合逻辑到底是属于哪个时钟域的，就没那么容易了，因为组合逻辑本身并不直接受时钟信号的控制。因此，要想判断组合逻辑的归属，需要分析它的所有输入：如果一个组合逻辑的所有输入都来自同一个时钟域内的寄存器的输出，那么该组合逻辑输出的变化频率必然也会跟随该时钟的脉搏跳动，所以可以确定该组合逻辑属于该时钟域；反之，则该组合逻辑中存在异步或者跨时钟域问题，它不属于任何一个时钟域。

时钟树简介

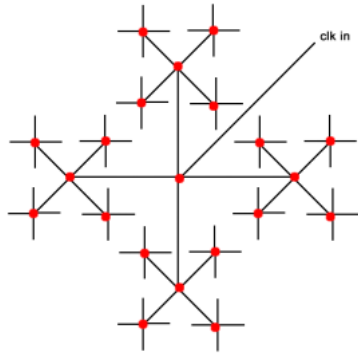
既然是“势力范围”，那么不同时钟信号的时钟域大小就有可能不同，这就好比世界上有很多个国家，可每个国家的领土大小、人口数量却各不相同。如果一个国家只有一个人，那么执政者只用严于律己就好，因为一人吃饱全家不饿；如果一个国家只有10个人，那么执政者只需要管理好自己以及其余9个人即可；如果人再多些，例如一个国家有10亿人，那么执政者就没有那个精力来挨个管了，于是就需要有省一级、市一级、区一级、乡一级等等的领导班子来组成所谓的公务员团体，来协助其进行管理。因此，对于“势力范围”大小不同的各个时钟域，时钟的管理方法也必然不同，只不过这次的帮手不是公务员，而是时钟树！（时钟树其实就是FPGA内部的时钟网络资源。）

那么，时钟树是怎么帮助时钟来“管理”时钟域的呢？

首先，如果一个时钟域包含了10000个触发器，那么该时钟信号必须要能够同时连接到这10000个触发器的时钟输入端，而如此大的信号扇出要怎么实现呢？不用担心，时钟树可以做到这么大的扇出。

其次，为了达到更好的时序指标，我们总希望时钟信号尽可能同时的到达其时钟域内各个触发器的时钟输入端，那么如此苛刻的要求又要怎么实现呢？不用担心，时钟树可以保证时钟信号到达时钟域内不同触发器的时间差最小。

由此可见，时钟树资源对时钟信号的重要性，我们可以形象的通过如下图示来理解时钟树的工作原理：



上图即为一个时钟树的原理结构图，时钟信号由最中间的树根引入时钟树，并经由第一级缓存扇出若干个分别送往不同的区域；在各个区域中，再采用同样的方式继续扇出并送往不同的子区域；如此往复，直到时钟信号到达最基本的触发器等资源。由此可见，时钟树主要利用多级扇出的形式来达到最终的高扇出结果；而通过控制每级扇出后的布线长度尽量一致来保证时钟信号能够尽可能同时的到达各个触发器，这就是时钟树的基本工作原理。注意，如果两个时钟域分别对应一个时钟信号的上升沿和下降沿，则它们可以共用一个时钟树。

最后，需要澄清一个概念，那就是时钟树是保证时钟信号到达时钟域内不同触发器的时间差尽可能小的资源，而不是保证时钟信号到达触发器所消耗时间最短的资源。例如，时钟域内有3个触发器，如果利用时钟树资源，并假设0时刻是时钟信号的初始时刻，那么时钟信号到达这三个触发器的时刻可能为8ns、9ns、8.5ns，而如果不应用时钟树，时钟信号到达这三个触发器的时刻可能为2ns、6ns、11ns。

时钟树的分类

既然时钟树是帮助时钟来管理时钟域的，那么由于时钟域可能很大、也可能很小，因此时钟树也会有大有小。由于时钟树资源在FPGA芯片中属于比较稀有的资源，那么为FPGA设计的每个时钟信号选择合适的时钟树就显得十分重要。那么，按照时钟树的作用范围来看，FPGA内部的时钟树资源共有三种，即：全局时钟树、区域时钟树、IO时钟树，相关的具体说明可以回顾本书【知己知彼篇->FPGA内部资源介绍->时钟网络资源】小节。

正确的时钟使用方式

了解了时钟域的相关内容，那么在平时的FPGA设计工作当中，该怎样正确运用呢？这一小节，将为大家介绍一些正确的时钟使用方式。

什么情况下时钟应该“上树”？

如果一个时钟信号是为FPGA内部的一些逻辑资源提供“脉搏”的，那么强烈建议让该时钟上树；如果时钟信号的时钟域实在太小，例如仅控制若干个触发器的话，那么也许不利用时钟树，FPGA设计也可能通过时序分析，但是仍然建议使用时钟树；如果时钟信号的时钟域只包括一个触发器的话，那么也就不存在所谓的时间差了，此时就完全不需要时钟树；如果一个时钟信号仅仅是为FPGA外围的硬件电路提供时钟激励的，那么无论外部有多少个存储单元需要使用该时钟，都没有必要使用时钟树，因为FPGA内部的时钟树无法延伸到FPGA芯片外部，就好比美国公民不受中国法律约束一样。

如何选择时钟树？

如果需要使用时钟树，那么该为时钟选择哪一类时钟树呢？

也许你会想当然的觉得，时钟域大的，选择全局时钟树；时钟域较小的，选择区域时钟树；时钟域特别小的，选择IO时钟树。那么在这里，我只能告诉你，完全错误！

首先来看IO时钟树，IO时钟树只分布在FPGA的接口资源中，由于它们离IO管脚最近，所以可以协助FPGA完成一些较高速率的串行数据接收，在经过简单的串并转换之后，以比较低的速率将并行数据丢进FPGA芯片的内部，供其他资源使用。一般来说，每个IO BANK内部会有若干个IO时钟树的资源，因此，IO时钟树虽然覆盖范围小，但并不是为小规模时钟域量身定做的，因此FPGA内部的资源也无法使用该时钟树。

再来看全局时钟树，由于全局时钟树可以覆盖到整个FPGA芯片，因此全局时钟树的个数也十分有限，因此使用的时候

是一件很痛苦的事情，如果你硬着

点赞1

评论

分享

收藏12

举报

关注

一键三连

派上用场，那简直是浪费时间、白花心思。因此，全局时钟树这样的“黄金地段”，不能不用，不能滥用，而要充分利用！因此，在全局时钟树资源不紧缺的情况下，无论时钟域的大小，统一建议使用全局时钟树，因为这样也能够给编译器提供最大的布局布线自由度，从而让时序约束更容易实现。

最后来看区域时钟树，老实说，区域时钟树的覆盖范围也是相当大的，最大可能到FPGA芯片的几分之一，因此如果时钟域不是特别大的话，到底使用全局时钟树还是区域时钟树，其实没有一个确定的结论。不过如果不是全局时钟树资源不够用的话，一般不建议使用区域时钟树。当然了，使用区域时钟树可以让时钟域中资源的分布在物理上更加紧凑一些，并且有些功能是必须使用区域时钟树和IO时钟树配合来完成的，因此请注意相关功能的说明。

时钟信号怎么“上树”？

明确了该为时钟信号使用的时钟树种类，那么接下来就来看看具体的做法吧。

使用全局时钟树资源

方法一，通过正确的物理连接。

如果时钟信号是由FPGA芯片的外部产生，那么我们可以不通过编程就实现时钟树资源的分配。因为在FPGA芯片的外围管脚中，有一些专门为全局时钟设计的管脚，这点我们可以通过相应FPGA芯片的数据手册来确认，如果在制作电路板时，直接将外部时钟信号通过这些管脚接入FPGA内部，那么它将自动占有全局时钟树资源。当然了，这些管脚也可以接入普通的数据信号，编译器会对该管脚引入的信号在FPGA设计内部扮演的角色进行分析，如果发现其并没有作为时钟信号来使用，那么将不会为其分配时钟树资源。

方法二，通过恰当的代码描述。

如果很不巧，外部的时钟信号没有通过专用的全局时钟管脚连接到FPGA内部，又或者某一个时钟信号本身就是在FPGA内部产生的，例如FPGA内部PLL的输出，那么此时就需要通过编写程序来完成时钟的“上树”工作了。有些时候，即使不使用代码显式指定，编译器也会根据代码的分析结果，来为时钟信号分配全局时钟树资源，不过这种“靠天吃饭”的思想不可取，FPGA工程师一定要让FPGA芯片尽可能的处于自己的掌控之下，而不是编译器，因此强烈建议大家通过自己的代码来指明时钟树的使用。那么具体要怎么通过HDL代码来实现时钟树资源的分配呢？使用原语。更多关于原语的介绍请参阅【本篇->编程思路->原语的使用】章节，在这里，我们只简单介绍一下如何使用原语来完成时钟信号的“上树”。

由于原语是跟FPGA芯片的生产厂商息息相关的，因此同一个功能的原语在不同的编译器中的名称很可能大相径庭，例如，用于全局时钟树分配的最主要的原语，Xilinx公司叫它BUFG，而Altera公司却称其为global。在这里，我们将以Xilinx公司的FPGA产品为例，来介绍代码的描述方法，其他公司的FPGA产品方法类似，只不过需要替换原语的名称罢了。

如果FPGA内部有一个名为innerClk的时钟信号，我们想为它分配一个全局时钟树，该怎么办呢？可以使用如下HDL描述：

```
1  -- VHDL example
2  signal globalClk : std_logic;
3  onTree: BUFG port map(0 => globalClk, I => innerClk);
```

```
// Verilog example
wire globalClk;
BUFG onTree(.O(globalClk), .I(innerClk));
```

```
1
2
3
4
5
6
7
```

按照上述HDL代码处理后，我们就可以在后续的逻辑功能中放心使用“上树”后的innerClk——globalClk了。

实际上，直接从外部全局时钟管脚引入的时钟信号，相当于在HDL代码中使用了IBUFG+BUFG原语。除此以外，如果希望多个时钟信号分享一个时钟树，也可以使用BUGMUX这个原语，相当于MUX+BUFG，例如，希望当前FPGA设计的某一部分逻辑其时钟是可以在40MHz和60MHz之间切换的。

使用区域时钟树、IO时钟树资源

与全局时钟管脚类似，FPGA芯片的外围管脚中也有专门为区域时钟和IO时钟设计的专用管脚，但是，光将时钟信号连接到这些管脚上，还并不一定能完成相应时钟树的使用，还必须要代码中显式的进行描述才行。以Xilinx公司为例，使用原语BUFIO，将会为这些专用管脚上的信号分配IO时钟树资源，使用原语BUFR，将会为这些专用管脚上的信号分配区域时钟树资源。由于区域时钟常配合IO时钟完成串并转换，因此BUFR通常还具有神奇的分频功能。最后，由于这两个时钟树的覆盖范围并不是整个FPGA芯片，所以在进行HDL代码编写时，也请注意资源的使用。

被“拉下树”的时钟信号

已经“上树”的时钟信号，如果不加小心，也可能被“拉下树”，因此在进行HDL代码编写的时候，一定要避免出现这种情况。那么是什么导致时钟信号脱离了时钟树呢？原因是这样的：通过前面小节介绍，我们知道时钟树是由若干级缓冲器再加一些近似等长的连线组成的，这也就是说时钟树仅能对时钟信号起到一个基本的传递作用，除此以外别无它用。因此，凡是想对时钟树上的时钟信号进行任何逻辑操作，来生成一个新的信号，那么新的信号就不再位于时钟树上了（注意，原始的时钟信号仍然在树上）。如果希望新的信号仍然作为时钟来驱动一些逻辑，那么必须重新调用相应原语来让新的时钟信号获得空闲的时钟树资源，所以，之前介绍的FPGA内部的再生时钟、门控时钟、行波时钟，如果需要使用的话，一定要先使用原语为它们分配好时钟树资源。

下例中，更加形象的说明了原始时钟信号被“拉下树”和再次“上树”的过程：

```
1  -- VHDL example
2  -- gClkOnTreeA is on the clock tree
3  midClk0 <= not gClkOnTreeA; -- midClk0 is not on the clock tree;
4  midClk1 <= en and gClkOnTreeA; -- midClk1 is not on the clock tree;
5  -- gClkOnTreeB is on the clock tree
6  reOnTree0: BUFG port map (0 => gClkOnTreeB, I => midClk0);
7  -- gClkOnTreeC is on the clock tree
8  reOnTree1: BUFG port map (0 => gClkOnTreeC, I => midClk1);
```

异步fifo跨时钟域处理 11-30
使用FPGA内部FIFO做跨时钟域的信息处理，避免亚稳态的传播。

FPGA之道（47）时钟及时钟域 Reborn Lee 792
本文节选自《FPGA之道》，通过作者的讲解，一起看看时钟和时钟域的相关问题，下一篇博客再...

优质评论可以帮助作者获得更高权重 抢沙发 评论

异步时钟域同步化处理方法 09-02
讲述了几种使用的异步时钟域同步化处理的方法

电子时钟微机原理课程设计报告 07-03
一、课程设计目的和意义 掌握8255、8259、8253芯片使用方法和编程方法，通过本次课程设计，...

...频率同步、同相位时钟、同源时钟、同时钟域时钟和异... 10-28
但是从波形上看两个时钟频率一样,跳变沿完全对称的不一定是同时钟域时钟,如两个PLL输出的看...

【FPGA】时钟域问题简介_Reborn Lee 12-1
上篇博文:时钟简介,简单的介绍了下时钟,这篇博文讲下时钟域的相关问题。时钟域的概念随着FPG...

内存spd引脚定义图 05-20
内存spd引脚定义图,有各引脚的功能,在内存的2号和3号脚基本不用。

一维热传导方程数值解法及matlab实现 03-25
含matlab程序，个人感觉很有帮助，在

点赞1 评论 分享 收藏12 举报 关注 一键三连