# 静态时序分析基础

# 参考

1、讲的很明白的博客：<https://blog.csdn.net/weixin_43343190/article/details/ 82960601>🡪[深入理解静态时序分析](#_深入理解静态时序分析)(本文位置)

2、好用的图：<https://wenku.baidu.com/view/380871878762caaedd33d487. html>

3、很好的练手题：<https://blog.csdn.net/Reborn_Lee/article/details/100049997>

🡪[例题一：](#_例题一：) [例题二：](#_例题二：) [例题三：](#_例题三：) [例题四：](#_例题四：) [例题五：](#_例题五：)

4、相关参考书：《FPGA原理和结构》第一章，《CMOS超大规模集成电路设计》第十章，《硬件架构的艺术》

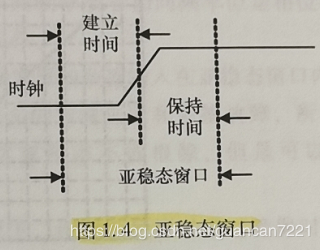
# 概念先导

**建立时间（Setup Time）：**在时钟脉冲到来前，输入数据需要保持稳定的时间。

**保持时间（Hold Time）：**在时钟脉冲到达后，输入数据需要继续保持的时间。

**亚稳态（Metastable）：**亚稳态是由于违背了触发器的建立和保持时间而产生的。信号如果在亚稳态窗口里发生变化，那输出将是未知的或称为“亚稳”的，这种有害状态的传播叫做亚稳态。触发器的输出会因此而产生毛刺，或者暂时保持在不稳定的状态而且需要较长时间才能回到稳定状态。（硬件架构的艺术，p1）

**亚稳态窗口：**输入信号应在亚稳态窗口中保持不变，否则输出可能变成亚稳态**。**



**静态时序分析（Static Timing Analysis，STA）**：静态时序分析是一种重要的逻辑验证方法，一般采用穷举分析方法来提取出整个电路存在的所有时序路径，计算信号在这些路径上的传播延时,检查信号的建立和保持时间是否满足时序要求,通过对最大路径延时和最小路径延时的分析,找出违背时序约束的错误。它关注的是时序间的相对关系，而不是逻辑功能。

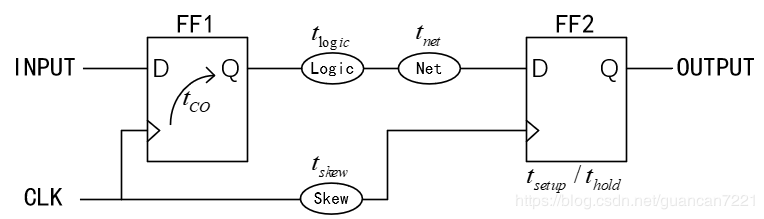
**时序路径（Timing Path）：**数据传输通路。

**关键路径：**常指同步电路中，组合逻辑延时最大的路径。

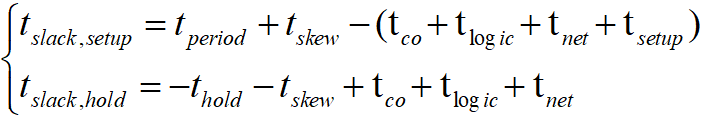
# 静态时序分析基础

做时序分析题之前先默念：“**时序设计的实质就是满足每一个触发器的建立/保持时间的要求**”

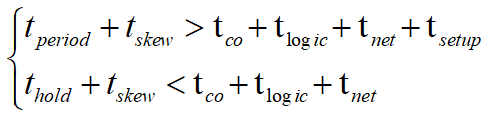
同步逻辑时延模型（简化）如下：



有如下十分全面而好使的公式：



对于有意义的时序约束，建立时间余量和保持时间余量都应大于0，因此可以推出如下衍生公式：（重点公式）



有了上边两组公式就能应对大多数时序分析题了（？）

公式中变量解释如下：

t slack,setup：触发器建立时间余量，对于有意义的时序约束，应大于0。

t slack,hold：触发器保持时间余量，对于有意义的时序约束，应大于0。

t setup：触发器的建立时间。

t hold：触发器保持时间。

t period：时钟周期。

t co ：触发器开始采样瞬间到采样数据开始在Q端输出的时间。

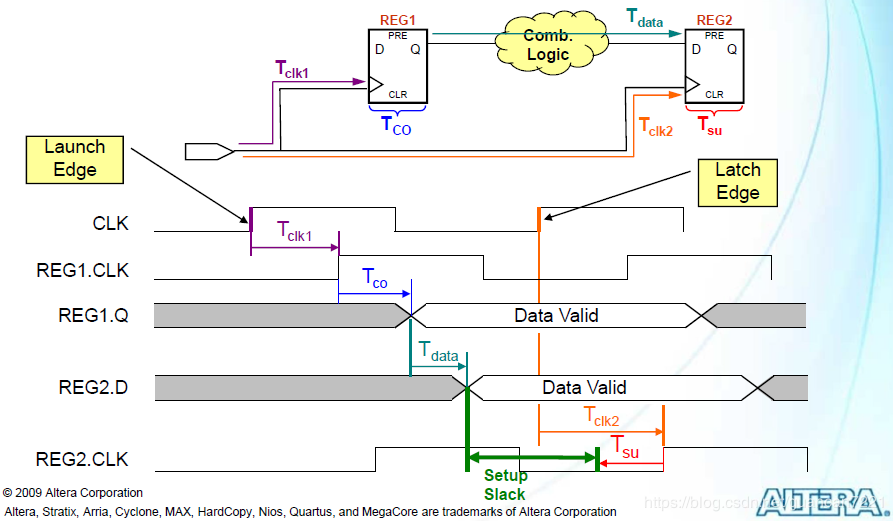
t logic：数据通过组合逻辑的延时。

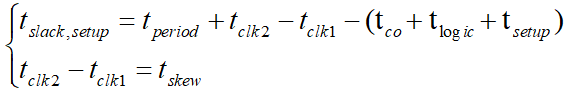
t net ：走线延时。

t skew ：时钟偏移，偏移的原因是因为时钟到达前后两个D触发器的路线不是一样长。

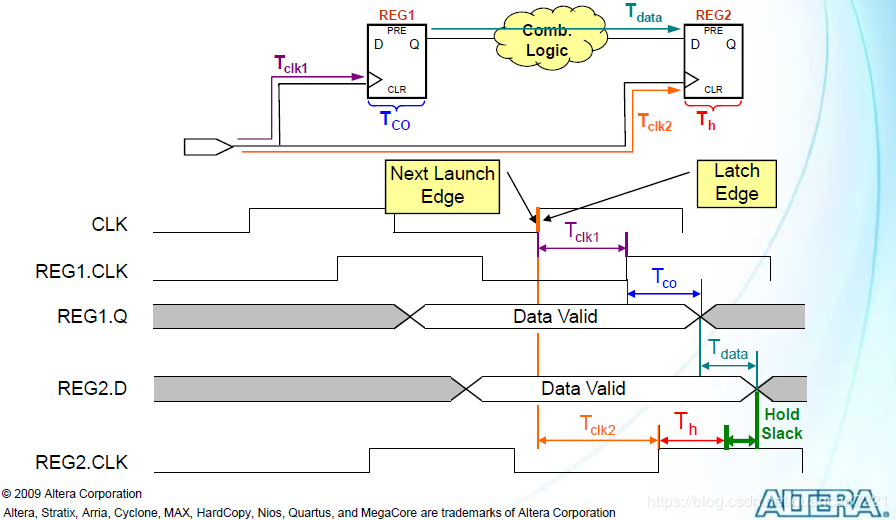
公式意义解释：

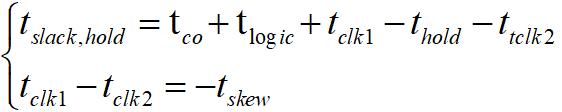
对于建立时间余量（图中忽略了走线延迟，但存在时钟偏移）：





对于保持时间余量，（图中忽略了走线延迟，但存在时钟偏移）：



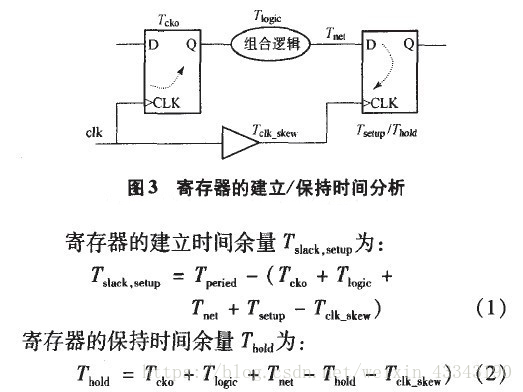


# 结语

到此可以去趁热去博客3做一做相关的题。此外在推公式时，有一些先导概念被我省略了，可以去参考2查看。

# 深入理解静态时序分析

我们的分析从下图开始，下图是常用的静态分析结构图，一开始看不懂公式不要紧，因为我会在后面给以非常简单的解释：



这两个公式是一个非常全面的，准确的关于建立时间和保持时间的公式。其中Tperiod为时钟周期；Tcko为D触发器开始采样瞬间到D触发器采样的数据开始输出的时间；Tlogic为中间的组合逻辑的延时；Tnet为走线的延时；Tsetup为D触发器的建立时间；Tclk\_skew为时钟偏移，偏移的原因是因为时钟到达前后两个D触发器的路线不是一样长。

这里我们来做如下转化：

因为对于有意义的时序约束，建立时间余量Tslack,setup和保持时间余量Thold都要大于0才行，所以对于时序约束的要求其实等价于：

**Tperiod>Tcko+Tlogic+Tnet+Tsetup-Tclk\_skew (1)**

**Tcko+Tlogic+Tnet>Thold+Tclk\_skew (2)**

之前说了，这两个公式是最全面的，而实际上，大部分教材没讲这么深，他们对于一些不那么重要的延时没有考虑，所以就导致不同的教材说法不一。这里，为了得到更加简单的理解，我们按照常规，忽略两项Tnet和Tclk\_skew。原因在于Tnet通常太小，而Tclk\_skew比较不那么初级。简化后如下：

**Tperiod>Tcko+Tlogic+Tsetup (3)**

**Tcko+Tlogic>Thold (4)**

简单多了吧！但是你能看出这两个公式的含义吗？其实（3）式比较好理解，意思是数据从第一个触发器采样时刻传到第二个触发器采样时刻，不能超过一个时钟周期啊！假如数据传输超过一个时钟周期，那么就会导致第二个触发器开始采样的时候，想要的数据还没有传过来呢！那么（4）式又如何理解呢？老实说，一般人一眼看不出来。

我们对于（4）式两边同时加上Tsetup，得到（5）：

**Tcko+Tlogic+Tsetup>Thold+Tsetup (5)**

结合（3）式和（5）式，我们得到如下的式子：

**Thold+Tsetup <Tcko+Tlogic+Tsetup< Tperiod (6)**

这个式子就是那个可以让我们看出规律的式子。也是可以看出静态时序分析**本质**的式子。

**Tcko+Tlogic+Tsetup**是指数据从第一级触发器采样瞬间开始，传输到第二级触发器并被采样的传输延时。我们简称为数据传输延时。下面讲述（6）式两端的含义。

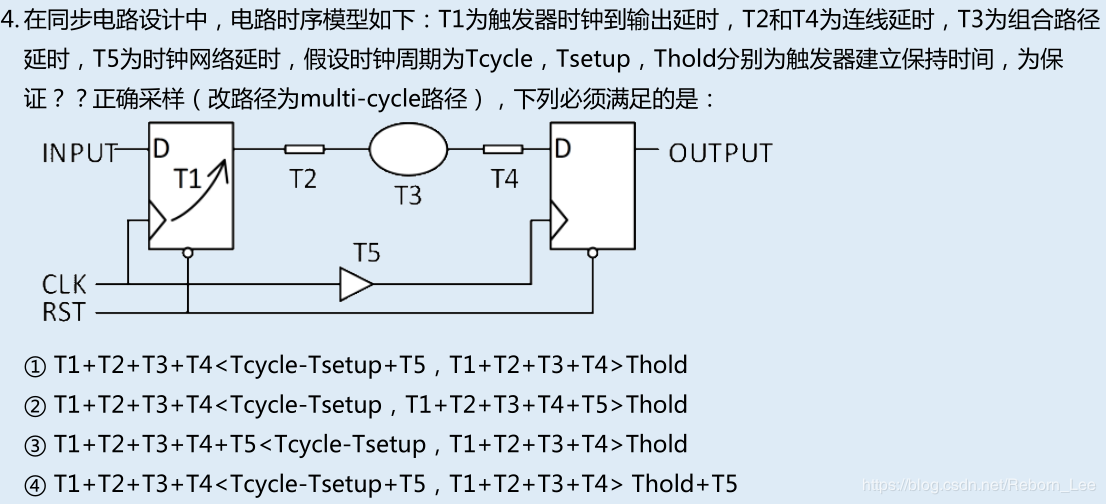
**Tcko+Tlogic+Tsetup< Tperiod** ：约定数据传输延时不能太大，如果太大（超过一个时钟周期），那么第二级触发器就会在采样的时刻发现数据还没有到来。

**Thold+Tsetup <Tcko+Tlogic+Tsetup**：约定数据传输延时不能太小。这就奇怪了，数据传得太快大家都知道不好，难道传得太慢也不行吗？是的，不行！Thold+Tsetup是一个触发器的采样窗口时间，我们知道，D触发器并不是绝对的瞬间采样，它不可能那么理想。在D触发器采样的瞬间，在这瞬间之前Tsetup时间之内，或者这瞬间之后Thold时间之内，如果输入端口发生变化，那么D触发器就会处于亚稳态。所以采样是有窗口的，我们把Thold+Tsetup的时间宽度叫做触发器的采样窗口，在窗口期内，D触发器是脆弱的，对毛刺没有免疫力的。假如数据传输延时特别小，那么就会发现，当第二级触发器开始采样的时候，第一级触发器的窗口期还没有结束！也就是说，如果这个时候输入端数据有变化，那么不仅第一级触发器处于亚稳态，第二级触发器也将处于亚稳态！

综上，我们就可以知道，数据传输延时既不能太大以至于超过一个时钟周期，也不能太小以至于小于触发器采样窗口（亚稳态窗口）的宽度。这就是静态时序分析的终极内涵。有了这个，就不需要再记任何公式了。

# 例题一：

有了这些基础，我们从最简单的考题分析：[IC/FPGA笔试题分析（五）](https://blog.csdn.net/Reborn_Lee/article/details/98524747)



这个题目堪称经典中的经典，因为从这个题目是标准的建立时间和保持时间考题，并从中可以总结出系统最大时钟频率以及建立时间和保持时间需要满足的公式。

我们知道系统周期需要满足的条件是：

Tcycle + Tskew > Tco + Tgate + Tsu;

代入上面的条件，得知

Tcycle + T5 > T1 + T2 + T3 + T4 + Tsetup

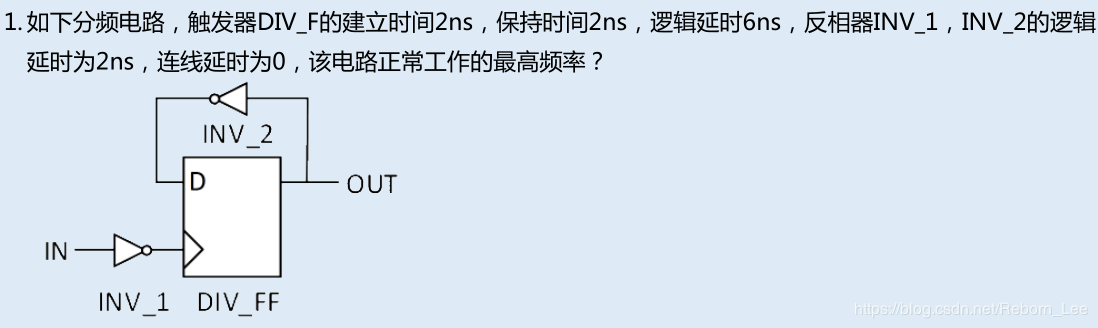
上面的时钟路径是存在偏斜的，而且是正时钟偏斜，则对于保持时间的满足更加的困难，需要满足：

Thold + Tskew < Tc0 + Tgate

代入上面的条件，得： Thold + T5 < T1 + T2 + T3 + T4

# 例题二：

在上题的基础上，你也许会分析触发器的建立时间和保持时间应该满足的关系了，可是考题也许会变了法的考你，如下：



求系统的最高频率？

就一个触发器，我怎么办呢？

仔细看来，这是一个触发器到自身的反馈，可以看做两个触发器之间进行数据传输，其实也就是如此。

需要明白的是，由于时钟到达这个触发器的时间一致，所以不存在时钟偏斜。

给出系统周期满足的关系：

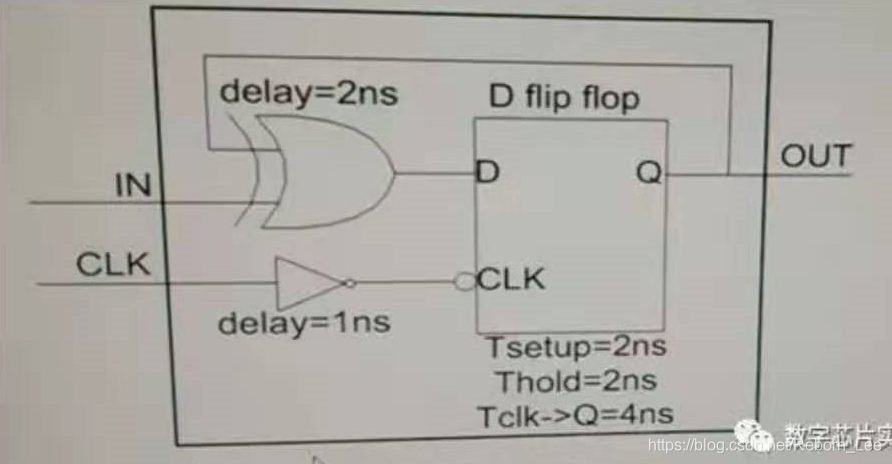
Tcycle > Tco + Tgate + Tsu = 2 + 6 + 2= 10ns

所以时钟最大频率可以为100MHz。

# 例题三：

如果都是分析触发器的建立时间和保持时间就好了，可惜不是这样的，套路来了：

如下图，将框内的电路作为一个寄存器，那么其有效的建立时间和保持时间是多少？



这个题目猛地一看就让人懵逼了，我还曾因项目调试不顺利，遇到这个题目，心态炸裂过。。。（年轻）

求框内电路的有效建立时间和保持时间，给出了框内触发器的建立时间和保持时间以及其他一些逻辑的延迟，我们就以触发器的建立时间和保持时间为准来倒逼整个电路的建立时间和保持时间，一切还都是为了让触发器满足时序要求。

**有效建立时间分析：**

假设电路的有效Setup为Tsetup\_valid：

对于D触发器而言，其本身的建立时间是2ns，也就是说数据必须在时钟有效沿到达之前2ns保持稳定，这样到达D端后就一定是稳定的数据了。

这个电路的数据来自于IN，时钟来自于CLK；

考虑时钟路径延迟影响：

时钟CLK要早于触发器的时钟1ns到达，因此对于D触发器建立时间的满足是有害的，电路有效建立时间

Tsetup\_valid = Tsetup - 1ns = 1ns（因为数据需要提前1ns稳定下来）

考虑数据路径延迟影响：

Tsetup\_valid = Tsetup - 1ns + 2ns = 3ns；（经过组合逻辑后的数据需要在时钟有效沿之前Tsetup时间稳定下来）

**有效保持时间分析：**

和建立时间分析套路一致，对于D触发器而言，数据需要在时钟有效沿到来之后保持Thold时间。

考虑时钟延迟的影响：

考虑到电路时钟对于触发器时钟早到1ns，所以电路有效保持时间Thold\_valid = Thold + 1ns = 3ns;

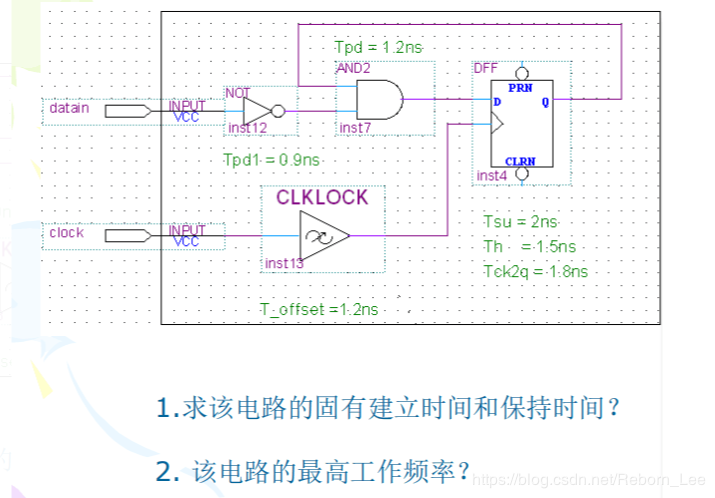
考虑路径延迟影响：

数据需要经过一段组合逻辑之后才能保持稳定，因此电路的有效保持时间为：

Thold\_valid = Thold + 1ns - 2ns = 1ns。

# 例题四：

下面这个题目和上面题目相差无几：



1. 求该电路的固有建立时间和保持时间？
2. 该电路的最高工作频率？

求电路的固有建立时间和保持时间和上一题的有效建立时间和保持时间是一个意思的。

因此固有建立时间为Tsu\_valid = Tsu - T\_offset + Tpd + Tpd1 = 2 - 1.2 + 1.2 + 0.9 = 2.9ns

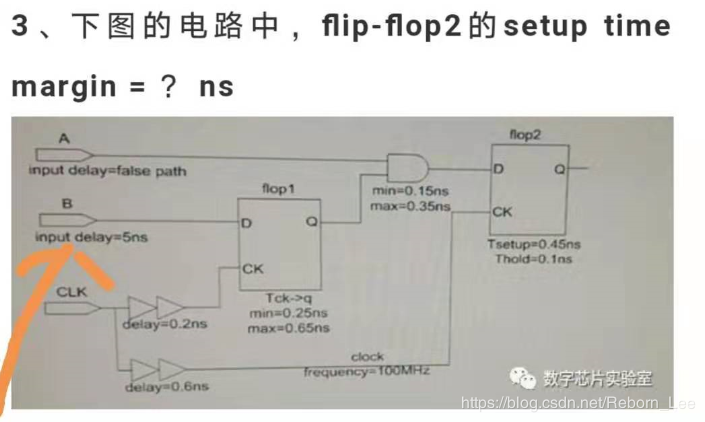
固有保持时间为：Th\_valid = Th + T\_offset - Tpd - Tpd1 = 1.5 + 1.2 - 1.2 - 0.9 = 0.6ns.

而系统的最高频率呢？

先求系统的最小周期，考虑两个触发器之间的路径：

Tmin = Tco + Tpd + Tsu = 1.8 + 1.2 + 2 = 5ns,那么系统最高频率为200MHz。

# 例题五：



这个题目让求setup time margin，意思大概就是建立时间裕量，就是系统周期减去Tco，Tgate以及Tsu之后还可以有多少裕量，那，Tco，Tgate以及Tsu当然要用最大的来代入，因为要保证系统在最恶劣的情况下，能有多少裕量。

因此：

Tmargin = 10ns + 0.6ns - 0.2ns - 0.65ns - 0.35ns - 0.45ns = 8.95ns

其中0.6ns - 0.2ns表示的是时钟偏斜量，可见是时钟整偏斜，有利于时钟裕量。