**静态时序分析圣经翻译计划——第十章：鲁棒性检查 （下）**



[**赵俊军**](https://www.zhihu.com/people/zhao-jun-jun-19)

[**​**](https://www.zhihu.com/question/48510028)

上海交通大学 电子与通信工程硕士在读

**[科学求真](https://www.zhihu.com/xen/market/ecom-page/1309219725549920256)**

[赢 10 万奖金 · 院士面对面](https://www.zhihu.com/xen/market/ecom-page/1309219725549920256)

23 人赞同了该文章

**10.9 统计静态时序分析**

到目前为止介绍的静态时序分析技术是确定性的，因为分析基于的是设计中所有时序弧的固定延迟。每个时序弧的延迟都是根据工作条件以及工艺和互连模型计算得出的，尽管可能存在多个模式和多个角，但给定情况下的时序路径延迟是可以明确获得的。

实际上，执行STA时通常使用的工艺和工作条件的最差情况（WCS）或最佳情况（BCF）对应于极端的3σ角。时序库基于的是代工厂提供的带有工作条件的工艺角模型，这些条件对应于单元时序值的不同角。例如，使用快速工艺模型、最高电源和最低温度来表征最佳情况的快速时序库。

**10.9.1 工艺和互连走线变化**

**全局工艺变化**

全局工艺变化（global process variation），也称为芯片间器件变化（inter-die device variations），是指影响芯片（或晶圆）上所有器件的工艺参数变化，参见图10-24。这表明芯片上的所有器件都会受到这些工艺变化的影响，芯片上的每个器件都会是slow或是fast的，或者介于两者之间。 因此，通过全局工艺参数建模的变化旨在捕获芯片与芯片之间的变化。

图10-24

图10-25中显示了全局参数值（例如g\_par1）的变化。例如，参数g\_par1可以对应于标准NMOS器件的IDSsat（器件饱和电流）。由于这是一个全局参数，因此芯片上所有单元实例中的所有NMOS器件将对应于相同的g\_par1值。可以有如下选择：所有单元实例的g\_par1的变化是完全相关的，或者芯片上g\_par1的变化相互影响。注意，可能还存在其它全局参数（g\_par2，...），其可能可以对PMOS器件饱和电流和其它相关变量建模。

图10-25

不同的全局参数（g\_par1，g\_par2，...）之间是不相关的。不同全局参数的变化是不会相互影响的，这意味着g\_par1和g\_par2参数彼此将独立地变化。在一块芯片上，g\_par1可能处于最大值，而g\_par2可能处于最小值。

在确定性（即非统计性）分析中，慢速工艺模型可能对应于芯片间变化的+ 3σ角的条件。类似地，快速工艺模型可能对应于芯片间变化的-3σ角的条件。

**局部工艺变化**

局部工艺变化（local process variation），也称为芯片内器件变化，是指工艺参数的变化，这些变化可以在给定芯片上影响不同器件，参见图10-26。这意味着并排放置在同一芯片上的相同器件可能具有不同的行为。由局部工艺变化建模的变化旨在捕获芯片内的随机工艺变化。

图10-26

图10-27显示了局部工艺参数的变化。芯片上的局部参数变化不会相互影响，并且它们从一个单元实例到另一单元实例的变化是不相关的。这意味着对于同一芯片上的不同器件，局部参数可能具有不同的值。例如，芯片上的不同NAND2单元实例可能会具有不同的局部工艺参数值。即使其它参数（例如输入压摆和输出负载）相同，这也可能导致同一NAND2单元的不同实例具有不同的延迟值。

图10-27

由全局和局部变化引起的NAND2单元延迟变化的示意图如图10-28所示。该图说明了全局参数变化比局部参数变化引起的延迟变化更大。

图10-28

局部工艺变化是打算在使用OCV建模的分析中捕获的变化之一，如10.1节中所述。由于统计时序模型通常包括局部工艺变化，因此使用统计时序模型进行的OCV分析不应在OCV设置中包括局部工艺变化。

**互连走线变化**

如10.8节中所述，实际上存在着各种互连角，它们代表影响互连电阻和电容值的每个金属层的参数变化。这些参数变化通常是金属和电介质的厚度以及影响各种金属层中金属走线的宽度和间距的金属刻蚀。通常，影响金属的参数会影响该金属层中所有走线的寄生参数，但对其它金属层中走线的寄生影响很小甚至是没有影响。

10.8节中介绍的互连角可用于对互连走线变化进行建模，以便所有金属层都对应到相同的互连角下。对互连走线变化进行统计建模时，每个金属层都可以独立地变化。统计方法会对互连走线空间中所有可能的变化组合进行建模，从而对仅通过在指定互连角下进行分析可能无法捕获的变化进行建模。例如，时钟树的发起路径可能在METAL2中，而时钟树的捕获路径可能在METAL3中。传统互连角下的时序分析会考虑各种角，这些角会同时改变所有金属层，因此无法对这种情况进行建模：METAL2在该角下延迟最大，而METAL3在该角下延迟最小。这种组合对应于路径建立时间检查的最差情况，并且只能通过对互连走线变化进行统计建模来捕获。

**10.9.2 统计分析**

**什么是SSTA？**

如果对单元时序模型和互连寄生进行统计建模，则上述对变化的建模是可行的。除延迟外，还对单元输入端的引脚电容值也进行了统计建模，这意味着时序模型是根据工艺参数（全局和局部）的均值和标准差来描述的，而互连电阻和电容是根据互连参数的平均值和标准差来描述的。延迟计算过程（在第5章中介绍的）会先获得每个时序弧（单元以及互连走线）的延迟，然后再用相对于各种参数的平均值和标准差来表示。因此，每个延迟都由平均值和N个标准差来表示（其中N是统计建模中独立工艺和互连参数的数量）。

由于通过各个时序弧的延迟是用统计形式表示的，因此统计静态时序分析SSTA（Statistical Static Timing Analysis）过程中会结合时序弧的延迟以获得路径延迟，该路径延迟同样会以统计形式表示（具有均值和标准差）。SSTA会根据独立的工艺和互连参数的标准差，来获得路径延迟的总体标准差。例如，考虑由两个时序弧组成的路径延迟，如图10-29所示。由于每个延迟分量都有其变化，因此根据变化是相关的还是不相关的，将对变化进行不同的组合。如果变化来自同一来源（例如，由相互影响的g\_par1参数引起），则路径延迟的σ仅等于（σ1 + σ2）。但是，如果变化是不相关的（例如由于l\_par1参数），那么路径延迟的σ等于  ，该值小于（σ1 + σ2）。当对局部（不相关的）工艺变化建模时，路径延迟σ较小的现象也称为各个延迟变化的统计抵消（statistical cancellation）。

图10-29

对于一个实际设计，既要建模相关变化，也要建模不相关变化，因此需要适当组合这两种类型变化的权重。

发起和捕获时钟的时钟路径延迟也以相同的统计形式表示。基于数据和时钟路径延迟，可以将裕量（slack）作为具有标准值和标准差的统计量。

假设使用正态分布，可以获得对应于（平均值mean +/- 3σ）的有效最小值和最大值。 （平均值mean -/ + 3σ）对应于图10-30中所示正态分布的0.135％和99.865％的分位数。0.135％的分位数意味着仅0.135％的结果分布小于此值（平均值mean-3σ）；同样，99.865％的分位数表示99.865％的结果小于此值，或仅0.135％（100％-99.865％）的结果大于此值（平均值mean+3σ）。有效的下限和上限在SSTA报告中称为分位数（quantile），设计人员可以选择分析中使用的分位数，例如0.5％或99.5％，对应于（平均值mean -/ + 2.576σ）。

图10-30

对于噪声和串扰分析（第6章），将使用相对于各种参数的平均值和标准差来对路径延迟和时间窗口进行统计建模。

根据路径裕量分布，SSTA会报告每个路径裕量的平均值、标准差和分位数，从而可以根据所需的统计置信度（confidence）来判断时序是通过还是违例。

**统计时序库**

在SSTA方法中，标准单元库（以及设计中使用的其它单元库）提供了各种环境条件下的时序模型。例如，在最小Vdd和高温角下进行的分析利用了在此条件下表征的库，但对工艺参数进行了统计建模。该库包含了用于标准参数值以及参数变化的时序模型。对于N个工艺参数，在0.9V电源和125°C条件下表征的统计时序库可能包括以下内容：

● 具有标准工艺参数的时序模型

● 参数i为（标准值+1σ）的时序模型，其他参数保持为标准值

● 参数i为（标准值-1σ）的时序模型，其他参数保持为标准值

对于仅具有两个独立工艺参数的简化情况示例，时序模型是以标准参数值以及参数值的变化来表征的，如图10-31所示。

图10-31

**统计互连走线变化**

每个金属层有三个独立的参数：

● 金属刻蚀：这可以控制金属宽度以及与相邻导体的间距。金属层中的大刻蚀减小了宽度（这增加了电阻）并增加了到相邻走线的间距（这减小了到相邻走线的耦合电容）。该参数表示为导体宽度的变化。

● 金属厚度：较厚的金属意味着与下面各层的电容更大。该参数表示为导体厚度的变化。

● IMD（金属间介电层）厚度：较大的IMD厚度会减少与下面各层的耦合。该参数表示为IMD厚度的变化。

**SSTA结果**

统计分析中输出的结果将根据平均值和角的有效值来提供路径的裕量。以下是用于建立时间检查（最大路径分析）的SSTA报告示例：



上面的报告显示，尽管时序路径的均值满足要求，但0.135％的分位数却有0.43ns的时序违例，时序路径裕量的分位数为-0.43ns。时序路径裕量的平均值为+ 0.86ns，标准差为0.43ns，这意味着+/- 2σ的分布结果满足要求。由于95.5％的分布落在2σ的变化范围内，这意味着只有2.275％的路径会出现时序违例（其余的2.275％分布具有较大的正裕量）。因此，把分位数设置为2.275％后可以使得裕量为0或没有时序违例。到达时间和路径裕量分布如下图10-32所示：

图10-32

请注意，以上报告是针对建立时间检查的，因此分位数那一列提供的是分位数上限（例如，路径延迟的+ 3σ值），保持时间检查时则会指定为分位数下限（例如-3σ值）。报告中的“sensitiv”列是指敏感度，即标准差与平均值的比值（表示为百分比）。就裕量而言，需要较小的灵敏度，这意味着即使平均值发生变化，以平均值通过的路径也会继续通过。“incr”列中指定了报告中该行的增量标准差。

使用单元和互连走线的统计模型，统计时序方法可以分析各种角条件下的设计，并分析由于工艺和互连参数变化而导致的情况。例如，在最差VT（电压和温度）情况下进行统计分析将分析整个全局工艺和互连走线空间。在最佳VT（电压和温度）情况下的另一种统计分析也将分析整个工艺和互连走线空间。这些分析可以与在最差PVT情况或最佳PVT情况下进行的传统分析进行对比，传统分析仅探讨工艺和互连空间中的单个点。

**10.10 时序违例路径**

在本节中，我们提供了一些示例，重点介绍了设计人员在STA调试期间需要重点关注的方面。其中一些示例仅包含STA报告中的相关片段。

**找不到路径**

如果有人试图获取路径报告而STA却报告找不到路径，或者它提供了路径报告但裕量是无限的，该怎么办？在这两种情况下，都可能是由于以下原因引起的：

● 时序路径被打断

● 路径不存在

● 这是一条伪路径

在每种情况下，都需要仔细检查约束条件，以识别导致路径阻塞的约束条件。一种“简单粗暴”的选择是删除所有伪路径和时序中断的设置，然后查看路径是否可以进行时序分析。（时序中断是在STA中中断时序弧，可通过使用7.10节中所述的**set\_disable\_timing**命令来实现）

**跨时钟域**

以下是一份路径报告的前面部分：



首先要注意的是，该路径从输入端口开始，到触发器的复位引脚结束，并且在该复位引脚上会进行恢复时间检查（图中表示为library recovery time）。接下来要注意的是，该路径跨越了两个不同的时钟域：用于发起输入数据的SYS\_IN\_CLK和用于恢复时间检查的PX9\_CLK。即使从时序报告中看不出来，但从设计知识上来看，也可以检查两个时钟是否完全异步，以及这两个时钟域之间的任何路径是否应申明为伪路径。

**反相衍生时钟**

创建衍生时钟时，需要谨慎使用**-invert**选项。如果使用**-invert**选项指定了衍生时钟，则STA会假定指定点处的衍生时钟属于指定的类型。但是根据逻辑，在设计中可能不会出现这种波形。STA通常会给出错误或警告消息，表明衍生时钟无法实现，但是它将继续进行分析并报告时序路径。

图10-33

考虑图10-33中示例，让我们在单元UCKBUF0的输出上使用**-invert**选项定义一个衍生时钟：

● **create\_clock** **-name** CLKM **-period** 10 **-waveform** {0 5} [**get\_ports** CLKM]

● **create\_generated\_clock** **-name** CLKGEN **-divide\_by** 1 **-invert** **-source** [**get\_ports** CLKM] [**get\_pins** UCKBUF0/C]

以下是基于上述约束的建立时间检查路径报告：



注意，STA会假设单元UCKBUF0的输出波形是时钟CLKM的反相时钟。因此上升沿为5ns，建立时钟捕获沿为15ns。除了时钟的上升沿是5ns而不是0ns之外，从时序报告中还看不出有什么问题。应该注意的是，由于错误是在发起时钟路径和捕获时钟路径的共同部分上，因此建立时间和保持时间检查是正确的。设计人员需要仔细分析和理解STA产生的警告和错误信息。

要注意的重要一点是，无论是否可实现，STA都会按照指定的方式创建衍生时钟。

现在，让我们尝试将带有**-invert**选项的衍生时钟移至单元UCKBUF1的输出处，看看会发生什么。

● **create\_clock** **-name** CLKM **-period** 10 **-waveform** {0 5} [**get\_ports** CLKM]

● **create\_generated\_clock** **-name** CLKGEN **-divide\_by** 1 **-invert** **-source** [**get\_ports** CLKM] [**get\_pins** UCKBUF1/C]

以下是建立时间检查的路径报告：



该路径看起来像半周期路径（half-cycle path），但这是不正确的，因为实际逻辑中的时钟路径上没有反相。STA将再次假定UCKBUF1 / C引脚上的时钟为**create\_Generated\_clock**命令中指定的时钟。因此，上升沿出现在5ns。捕获时钟为时钟CLKM，其下一个上升沿发生在10ns处。下面的保持时间检查路径报告也会出现与建立时间检查路径相似的异常情况：



通常，STA输出的结果将包含指示衍生时钟不可实现的错误或警告信息。调试此类不正确路径的最佳方法是在捕获触发器和发起触发器处绘制时钟波形，并尝试了解所示时钟沿是否确实有效。

**缺少虚拟时钟延迟**

考虑以下路径报告：



这是一条从输入端口开始的路径。请注意，起始的数据到达时间列为0。这表示在时钟VCLKM上未指定延迟，时钟VCLKM用于定义输入引脚RESET\_L上的输入到达时间。这很可能是一个虚拟时钟，这就是为什么会缺少数据到达时间的原因。

**IO延迟大**

当输入或输出路径有时序违例时，首先要检查的是时钟延迟，该延迟用作指定输入到达时间或输出所需时间的参考。这同样适用于前面所讲的例子。

要检查的第二件事是输入或输出延迟，即输入路径上的输入到达时间或输出路径上的输出所需时间。我们可能会发现这些数值对于目标频率是不现实的。输入到达时间通常是时序报告中数据路径的第一个值，而输出所需时间通常是时序报告中数据路径的最后一个值。



在这条时序违例的输入数据路径中，注意输入到达时间为14ns。在这种特殊情况下，输入到达时间约束中会存在一个错误，因为它太大了。

**IO缓冲器延迟不正确**

当路径经过输入缓冲器或输出缓冲器时，约束不正确可能会导致输入或输出缓冲器延迟值较大。在如下所示的情况中，请注意18ns这个较大的输出缓冲器延迟值，这是由于输出引脚上指定的负载值较大导致的。



**延迟值不正确**

当时序路径出现违例时，需要检查的一件事是发起时钟和捕获时钟的延迟是否合理，即确保这些时钟之间的偏斜（skew）在可接受的范围内。错误的延迟约束或生成时钟树时错误的时钟平衡（balancing）可能会导致发起时钟和捕获时钟路径间的较大偏斜，从而导致时序违例。

**半周期路径**

如前面的示例所述，设计人员需要检查违例路径的时钟域。随之而来的是，设计人员可能需要检查发起和捕获触发器的时钟沿。在某些情况下，可能会发现一个半周期路径（上升沿至下降沿路径或下降沿至上升沿路径）并且可能无法满足半周期路径的时序要求，又或者这些半周期路径不是真实存在的 。

**较大的延迟和过渡时间**

一个关键事项是要检查沿数据路径的延迟或过渡时间的值是否异常地大，其中一些原因可能是：

● 高扇出网络：未适当缓冲（buffered）的网络。

● 较长网络：需要在中间插入缓冲器的网络。

● 低强度单元：由于在设计中标有“don't touch”，因此未被更换的单元。

● 存储器路径：这些路径通常会由于存储器输入上的建立时间较长以及存储器输出上的输出延迟较大而导致时序违例。

**缺少多周期保持时间约束**

对于多周期建立时间约束N，常见的是缺少相应的多周期保持时间约束N-1。因此，这可能会导致工具在修复保持时间违例时插入大量不必要的延迟单元。

**未优化的路径**

STA违例也可能出现在尚未优化的路径上，可以通过检查数据路径来确定这种情况。是否存在延迟较大的单元？可以手动改善数据路径的时序吗？也许数据路径需要进一步优化，工具可能正在其它违例路径上工作。

**仍不满足时序的路径**

如果数据路径具有强驱动能力的单元，而该路径仍然时序违例，则需要检查布线延迟和线负载较高的引脚。这可能是下一个改进方法：也许可以将单元移动得更近一些，从而可以减少线负载和布线延迟。

**如果仍然不满足时序怎么办**

可以利用有用偏斜（useful skew）来帮助时序收敛。有用偏斜是指故意使时钟树失衡的地方，尤其是时序违例路径的发起时钟和捕获时钟路径，从而使时序在该路径上收敛。通常，这意味着可以延迟捕获时钟，以使捕获触发器的时钟在数据准备就绪稍后到达。当然，这假定了后续数据路径（即下一级触发器到触发器的数据路径）上有足够的裕量。

也可以尝试相反的操作，也就是说，可以使发起时钟路径更短，以便更早地发起来自发起触发器的数据，从而帮助满足建立时间要求。同样，只有在前一级触发器到触发器路径具有多余的裕量时，才能做到这一点。

有用偏斜技术不仅可用于修复建立时间违例，还可用于修复保持时间违例。此技术的一个缺点是，如果设计具有多种操作模式，则有用偏斜可能会在另一种模式下引起问题。

**10.11 验证时序约束**

随着芯片尺寸的增长，将越来越依赖静态时序分析所交付的时序。仅依赖STA的风险在于STA取决于时序约束的合理与否。因此，时序约束的验证成为重要的考虑因素。

**检查路径例外**

有一些工具可以根据设计的结构（网表）检查伪路径和多周期路径的有效性，这些工具会检查指定的伪路径或多周期路径约束是否有效。此外，这些工具还可能可以根据设计的结构生成缺少的伪路径和多周期路径约束。但是，这些工具生成的某些路径例外（path exception）也可能是无效的。这是因为这些工具通常使用形式验证技术（formal verification techniques）通过逻辑结构来确定伪路径或多周期路径，而设计人员对设计的功能行为有更深入的了解。因此，在接受并在STA中使用它们之前，设计人员需要检查工具生成的路径例外。可能还存在其它基于设计语义行为的路径例外，如果工具无法提取此类路径例外，设计者必须手动定义这些路径例外。

时序约束中最大的风险就是路径例外。因此，应在仔细分析设计后确定出伪路径和多周期路径。通常，相比于伪路径，最好使用多周期路径，这确保了该路径至少受到一定程度的约束。如果在已知或可预期的时间会对信号进行采样，则无论信号离时钟沿有多远，都应使用多周期路径，这样静态时序分析至少可以知道一些时序约束的信息。伪路径可能会导致时序优化工具完全忽略这些路径，而实际上，它们确实可能在经过大量时钟周期后才被采样。

**检查跨时钟域**

可用工具来确保设计中所有的跨时钟域均有效，这些工具还可以自动生成必要的伪路径约束。这样的工具也可能可以识别出非法（illegal）的跨时钟域，即数据在没有任何时钟同步逻辑的情况下跨越了两个不同时钟域。在这种情况下，这些工具可以提供在需要时自动插入合适的时钟同步逻辑的功能。请注意，并非所有跨异步时钟域都需要时钟同步器，应该取决于数据性质以及是否需要在下一个周期或几个周期后捕获数据。

使用STA检查跨异步时钟域的另一种方法是设置一个很大的时钟不确定度（uncertainty），该不确定度等于采样时钟的周期。这样可以确保至少存在一些时序违例行为，根据这些时序违例行为，可以确定适当的路径例外，或者将时钟同步逻辑添加到设计中。

**验证IO和时钟约束**

验证IO和时钟约束仍然是一个挑战，设计人员会经常进行时序仿真以检查设计中所有时钟的有效性。可以进行系统级时序仿真来验证IO时序，以确保芯片可以与其外围设备通信且没有任何时序问题。

第十章： 鲁棒性检查 （完）