**静态时序分析圣经翻译计划——第五章：延迟计算 （下）**

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

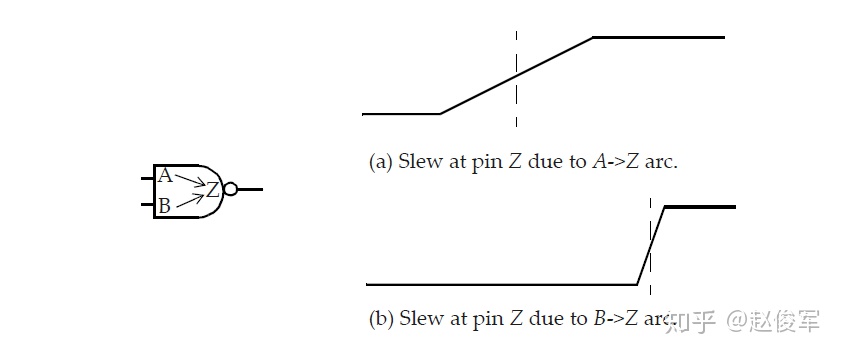
[**​**](https://www.zhihu.com/question/48510028)

上海交通大学 电子与通信工程硕士在读

22 人赞同了该文章

**5.4 压摆合并**

当多个压摆到达一个公共点时（例如在多输入单元的情况下）会发生什么呢？这种公共点称为压摆合并点（slew merge point）。 选择哪个压摆在压摆合并点处继续向下传播呢？考虑图5-10所示的2输入单元：

图5-10

由于引脚A上的信号改变，引脚Z上的压摆到达较早，但上升缓慢（压摆较小）；由于引脚B上的信号改变，引脚Z上的压摆到达较晚，但上升很快（压摆较大）。在压摆合并点（例如引脚Z），应选择哪个压摆进一步传播呢？取决于所执行的时序分析的类型（最大或最小时序路径分析），这些压摆值中的任何一个都可能是正确的。

进行最大时序路径分析时有两种可能性：

● 最差的压摆传播（Worst slew propagation）：此模式选择要在合并点传播最坏的压摆，这将是图5-10（a）中的压摆。对于通过引脚A-> Z的时序路径，此选择是准确的，但对于通过引脚B-> Z的任何时序路径都是悲观的。

● 最差的到达时间传播（Worst arrival propagation）：此模式选择要在合并点传播最差的到达时间，这对应于图5-10（b）中的压摆。在这种情况下选择的压摆对于通过引脚B-> Z的时序路径是准确的，但对于通过引脚A-> Z的时序路径是乐观的。

同样，进行最小时序路径分析时也有两种可能性：

● 最佳的压摆传播（Best slew propagation）：此模式选择要在合并点传播最佳的压摆，这就是图5-10（b）中的压摆。对于通过引脚B-> Z的时序路径，此选择是准确的，但对于通过引脚A> Z的任何时序路径，该选择的压摆值较小。对于经过A-> Z的路径，路径延迟小于实际值，因此对于最小时序路径分析是悲观的。

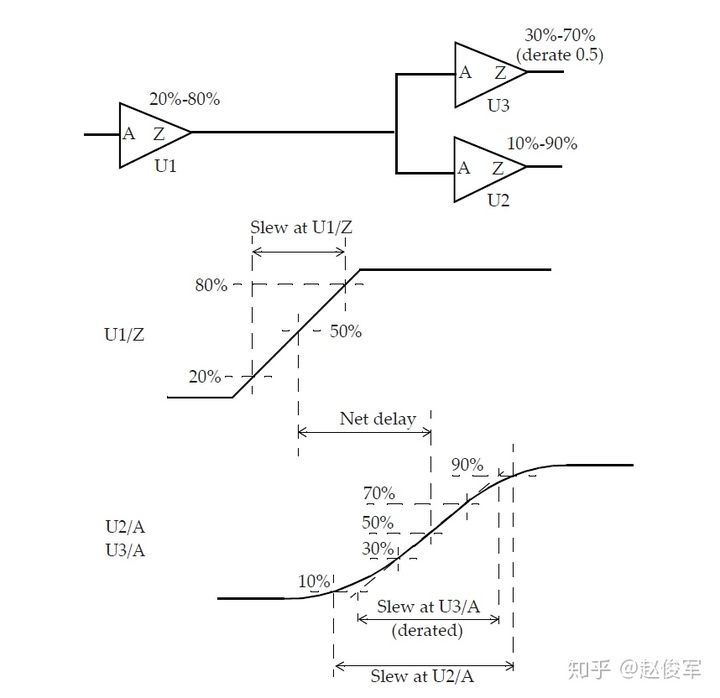
● 最佳的到达时间传播（Best arrival propagation）：此模式选择要在合并点传播最佳的到达时间，这对应于图5 10（a）中的压摆。在这种情况下，选择的压摆对于通过引脚A-> Z的时序路径是准确的，但大于通过引脚B-> Z的时序路径的实际值。对于经过B-> Z的路径，路径延迟大于实际值，因此对于最小时序路径分析是乐观的。

设计人员可以在静态时序分析环境之外执行延迟计算，以生成SDF文件。在这种情况下，延迟计算工具通常使用最差的压摆传播。生成的SDF文件足以用于最大时序路径分析，但对于最小时序路径分析可能是过于乐观的。

大多数静态时序分析工具均使用最差和最佳的压摆传播作为默认设置，因为它会保守地限制分析。但是，在分析特定路径时可以使用精确的压摆传播，精确的压摆传播需要在时序分析工具中启用一个选项。因此，重要的是要了解静态时序分析工具中默认使用哪种压摆传播模式，并清楚其可能过于悲观的情况。

**5.5 不同压摆阈值**

通常，库会指定在单元表征（characterization）期间使用的压摆（过渡时间）阈值。问题是，当具有一组压摆阈值的单元驱动其他具有不同压摆阈值设置的单元时，会发生什么呢？ 考虑图5-11中所示的情况，具有20%-80%压摆阈值的单元驱动两个扇出单元，其中一个具有10%-90%的压摆阈值，另一个具有30%-70%的压摆阈值且压摆降额系数为0.5。

图5-11

U1单元的压摆设置在单元库中定义如下：

**slew\_lower\_threshold\_pct\_rise：** 20.00

**slew\_upper\_threshold\_pct\_rise：**80.00

**slew\_derate\_from\_library：**1.00

**input\_threshold\_pct\_fall：**50.00

**output\_threshold\_pct\_fall：**50.00

**input\_threshold\_pct\_rise：**50.00

**output\_threshold\_pct\_rise：**50.00

**slew\_lower\_threshold\_pct\_fall：**20.00

**slew\_upper\_threshold\_pct\_fall：**80.00

U2单元的压摆设置在另一个单元库中定义如下：

**slew\_lower\_threshold\_pct\_rise：**10.00

**slew\_upper\_threshold\_pct\_rise：**90.00

**slew\_derate\_from\_library：**1.00

**slew\_lower\_threshold\_pct\_fall：**10.00

**slew\_upper\_threshold\_pct\_fall：**90.00

U3单元的压摆设置在另一个单元库中定义如下：

**slew\_lower\_threshold\_pct\_rise：**30.00

**slew\_upper\_threshold\_pct\_rise：**70.00

**slew\_derate\_from\_library：**0.5

**slew\_lower\_threshold\_pct\_fall：**30.00

**slew\_upper\_threshold\_pct\_fall：**70.00

上面仅展示了与U2和U3压摆相关的设置， 输入和输出阈值的延迟相关设置为50％，未在上面显示。延迟计算工具会根据连接到这个网络的单元的压摆阈值来计算过渡时间。图5-11显示了U1 / Z处的压摆如何与此引脚上的切换波形相对应，U1/ Z处的等效戴维宁源可用于获得扇出单元输入端的切换波形。延迟计算工具会根据U2 / A和U3 / A处的波形及其压摆阈值，计算出U2 / A和U3 / A处的压摆。请注意，U2 / A的压摆是基于10%-90%设置的，而U3 / A所用的压摆是基于30%-70%设置，然后根据库中指定的压摆降额（slew derate）系数0.5使用的。 此示例说明了如何根据切换波形和扇出单元的压摆阈值设置来计算扇出单元输入端的压摆。

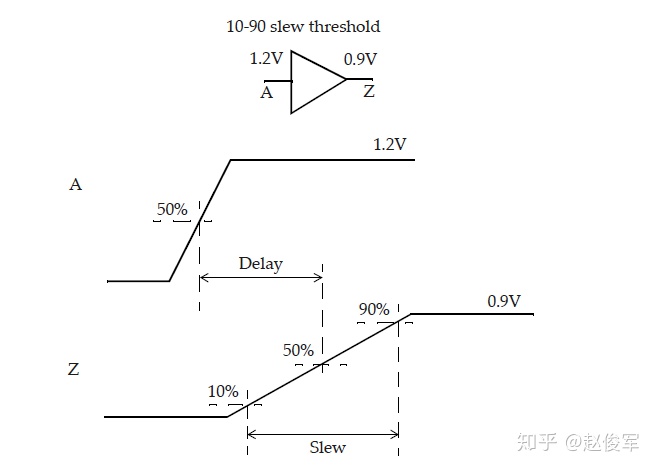
在可能不考虑互连电阻的预布局（pre-layout）设计阶段时，可以按以下方式计算具有不同阈值的网络处的压摆。例如，10%-90%压摆阈值和20%-80%压摆阈值之间的关系为：

● slew2080 / （0.8 - 0.2） = slew1090 / （0.9 - 0.1）

因此，10%-90%阈值测量点设置时的500ps压摆对应于20%-80%阈值测量点设置时（500ps \* 0.6）/ 0.8 = 375ps的压摆。类似地，20%-80%阈值测量点设置时的600ps压摆对应于10%-90%阈值测量点设置时（600ps \* 0.8）/ 0.6 = 800ps的压摆。

**5.6 不同电压域**

典型的设计中可能会对芯片的不同部分使用不同的电源。在这种情况下，在不同电压域之间的接口处应使用电平转换单元（level shifting cell）。电平转换单元在一个电压域接受输入，而在另一电压域提供输出。例如，一个标准单元输入可以为1.2V，其输出可以为较低的电压，如0.9V。下图5-12为一个示例：

图5-12

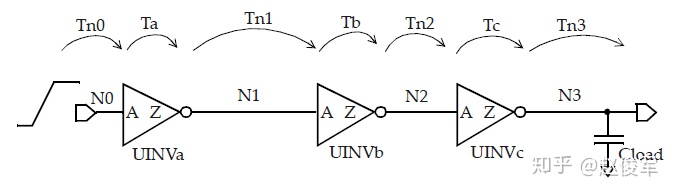
请注意，延迟是根据50％阈值点计算的。对于接口单元的不同引脚，这些点可能处于不同的电压。

**5.7 路径延迟计算**

一旦得到了每个时序弧的全部延迟，则可以将设计中各个单元的时序表示为时序图。通过组合逻辑单元的时序可以表示为从输入到输出的时序弧。类似地，互连线可用从源引脚到每个目的引脚的相应弧表示，表示为单独的时序弧。一旦整个设计由相应的时序弧标定（annotate）了，计算路径延迟就是将沿路径的所有网络和单元的时序弧相加起来即可。

**5.7.1 组合逻辑路径延迟**

考虑串联的三个反相器，如图5-13所示。在考虑从网络N0到网络N3的路径时，我们同时考虑了上升沿和下降沿路径，现假设网络N0处有上升沿。

图5-13

可以指定第一个反相器输入端的过渡时间（或压摆），若没有此类指定的话，就假定过渡时间为0（相当于理想情况）。输入UINVa / A处的过渡时间可通过使用上一节中指定的互连延迟模型来确定，同样，这个互连延迟模型也可用于确定网络N0的延迟Tn0。

根据UINVa输出处的RC负载，可以获得输出处UINVa/Z的有效电容。输入处UINVa/A的过渡时间和输出处UINVa/Z的等效有效负载则可用于获得单元输出下降延迟（output fall delay）。

等效的戴维宁电压源模型在引脚UINVa/Z上通过互连延迟模型可确定引脚UINVb/A上的过渡时间，互连延迟模型还用于确定网络N1上的延迟Tn1。

一旦知道了输入UINVb/A的过渡时间，就可以类似地计算通过UINVb的延迟。UINVb/Z处的RC互连以及引脚UINVc/A的引脚电容可用于确定N2处的有效负载。UINVb/A处的过渡时间可用于确定通过反相器UINVb的输出上升延迟（output rise delay），依此类推。

最后一级的负载由明确的负载说明来指定，如果没有指定，则仅使用网络N3的线负载。

上述分析假设网络N0为上升沿，对于网络N0的下降沿，可以进行类似的分析。因此，在这个简单的示例中，存在两条具有以下延迟的时序路径：

● Tfall = Tn0rise + Tafall + Tn1fall + Tbrise + Tn2rise + Tcfall + Tn3fall

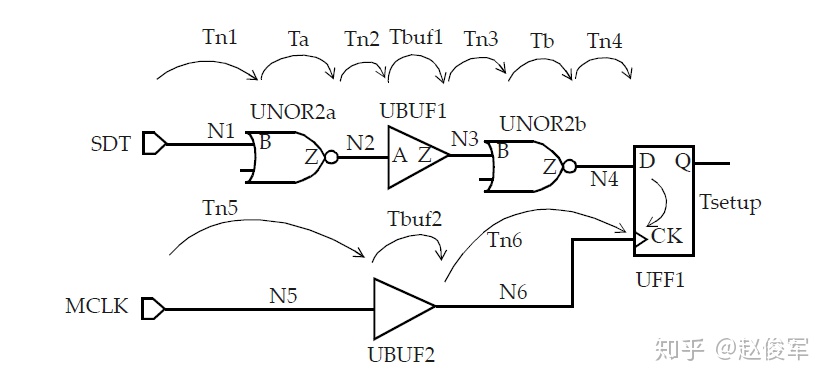
● Trise = Tn0fall + Tarise + Tn1rise + Tbfall + Tn2fall + Tcrise + Tn3rise

通常，由于驱动单元输出处的戴维宁电压源模型不同，通过互连线的上升和下降延迟可能会有所不同。

**5.7.2 到触发器的路径**

**输入到触发器路径**

考虑从输入SDT到触发器UFF1的时序路径，如图5-14所示：

图5-14

我们需要同时考虑上升沿和下降沿路径。对于输入SDT的上升沿，数据路径延迟为：

● Tn1rise + Tafall + Tn2fall + Tbuf1fall + Tn3fall + Tbrise + Tn4rise

同样，对于输入SDT的下降沿，数据路径延迟为：

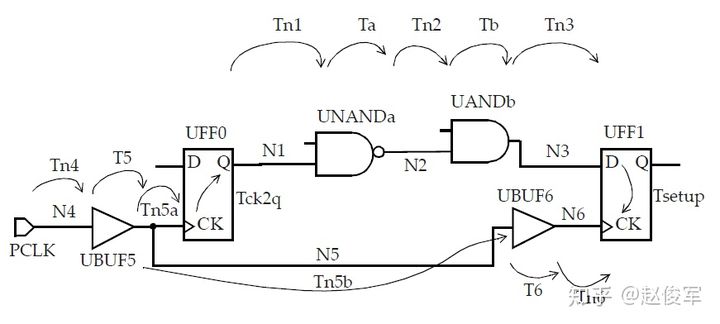
● Tn1fall + Tarise + Tn2rise + Tbuf1rise + Tn3rise + Tbfall + Tn4fall

输入MCLK上升沿的捕获（capture）时钟路径延迟为：

● Tn5rise + Tbuf2rise + Tn6rise

**触发器到触发器路径**

图5-15给出了两个触发器之间的数据路径和相应的时钟路径的示例：

图5-15

UFF0 / Q上升沿的数据路径延迟为：

● Tck2qrise + Tn1rise + Tafall + Tn2fall + Tbfall + Tn3fall

输入PCLK上升沿的发起（launch）时钟路径延迟为：

● Tn4rise + T5rise + Tn5arise

输入PCLK上升沿的捕获（capture）时钟路径延迟为：

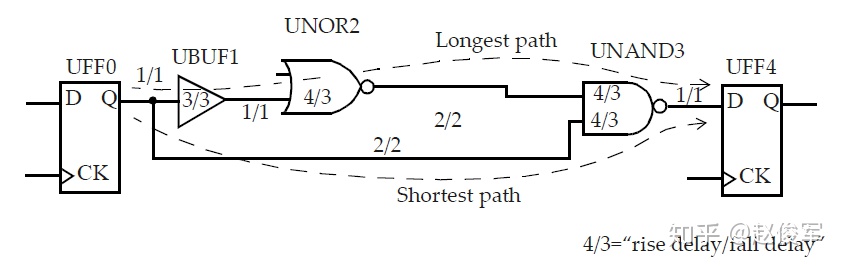
● Tn4rise + T5rise + Tn5brise + T6rise + Tn6rise

需要注意单元的单边性（unateness），因为边沿方向在通过单元时可能会改变（上升沿变下降沿，下降沿变上升沿）。

**5.7.3 多路径**

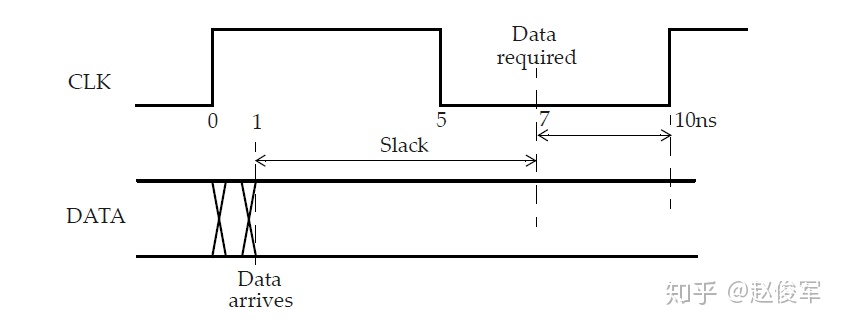
在任何两点之间，可以有很多路径。最长的路径是花费时间最多的路径，这也称为最差路径、较晚路径或最大路径。最短的路径是花费时间最少的路径，这也称为最佳路径、较早路径或最小路径。

请参见图5-16中时序弧的逻辑和延迟。两个触发器之间的最长路径是通过单元UBUF1、UNOR2和UNAND3，两个触发器之间的最短路径是通过单元UNAND3。

图5-16

**5.8 裕量计算**

裕量（Slack）是信号需要到达时间（Required Time）与实际到达时间（Arrival Time）之差。在图5-17中，要求数据在7ns时保持稳定才能满足建立时间（setup）要求。但是，数据在1ns时就已稳定。因此，裕量为6ns（= 7ns-1ns）。

图5-17

假设数据需要到达的时间是从捕获触发器的建立时间获得的，则计算过程如下：

● 需要到达时间Required\_Time = Tperiod - Tsetup = 10 - 3 = 7ns

● 实际到达时间Arrival\_Time = 1ns

● 裕量Slack = Required\_Time - Arrival\_Time = 7 - 1 = 6ns

同样，如果两个信号之间的偏斜（skew）要求为100ps，并且测得的偏斜为60ps，则偏斜的裕量为40ps（= 100ps-60ps）。

第五章： 延迟计算 （完）