

## 理解 FPGAs 中的亚稳态

这篇论文描述了在 FPGA 中的亚稳态，它是怎样产生的，还有它是怎样导致设计的失败。介绍了如何计算亚稳态的平均无故障时间(MTBF)，并且总结了各种器件和设计参数是如何影响平均无故障时间的。

### 介绍

亚稳态是当信号在无关的电路中或异步时钟域之间传输时导致数字器件包括 FPGA 系统失效的一种现象。这篇论文描述了 FPGA 中的亚稳态，解释了它是如何发生的，并讨论了它是如何导致设计失败的。

由亚稳态得出的平均无故障时间可以告诉设计者是否应该去降低出现失效的几率。这篇论文将告诉你如何由各种设计和器件参数计算出平均无故障时间，FPGA 供应商和设计者是如何提高平均无故障时间的。通过一些设计技巧和优化降低亚稳态造成的失效，系统可靠性是可以提高的。

### 什么是亚稳态

在所有的数字器件如 FPGA 中的寄存器都定义了一个信号时序要求，满足了这个要求寄存器才可以正确地在输入端获取(capture)数据在输出端产生数据。为了确保可靠的操作，输入信号在时钟沿之前必须稳定一段时间(寄存器建立时间  $T_{su}$ )并且在时钟沿之后稳定一段时间(寄存器保持时间  $T_h$ )，然后寄存器输出经过一个特定的时钟到输出延时(clock to output,  $T_{co}$ )后有效。如果一个数据信号在翻转中违反了一个寄存器的建立和保持时间的要求，寄存器的输出可能就会是亚稳态。在亚稳态中，寄存器的输出值在高和低之间徘徊一段时间，这就意味着输出翻转到一个确定的高或低的延时会超过固定的时钟到输出延时。

在同步系统中，输入信号必须总是满足寄存器时序要求，所以亚稳态不会发生。亚稳态问题通常发生在当一个信号在无关的线路中或异步时钟域中传输。在这种情况下设计者不能保证信号会满足建立时间( $T_{su}$ )和保持时间( $T_h$ )的要求，因为相对于目标时钟信号可以在任何时间到达。但是，不是每个违反寄存器建立保持时间的信号翻转都会导致亚稳态。一个寄存器进入亚稳态和从亚稳态进入稳态的时间依赖于制造商的制造工艺和使用环境。大部分情况下，寄存器会很快的进入一个定义的稳态中。

一个寄存器在时钟边沿取样一个数据信号可以形象地比喻成往一个小丘上抛球，如图 1 所示。这个小丘的每一侧代表一个稳态——信号翻转后的新旧数据值——小丘的顶部代表亚稳态，球可能会平衡在这儿，但在实际中它会慢慢地滑向(falls slightly)一侧。球落的离丘顶部越远，它就越快地到达在底部达到稳态。(The further the ball lands from the top of the hill, the faster it reaches a stable state at the bottom.)

如果一个数据信号在时钟边沿之后不超过最小保持时间之内翻转，类似于球落在小丘的“旧数据”端，对于这次时钟翻转输出信号仍旧保持原始值。当一个寄存器的数据输入在时钟沿的最小建立时间之前翻转并且保持超过最小建立时间，这类似于球落在小丘的

“新数据”端，输出很快的达到一个新问题以满足  $T_{co}$  的定义。然而当一个寄存器的数据输入违反了建立和保持时间，这类似于球落在了丘的顶部。如果球落在丘顶部附近，球将花费很长时间去到达底部，这将增大从时钟到输出的延迟并超过定义的  $T_{co}$ 。

图 1 亚稳态抛球图示

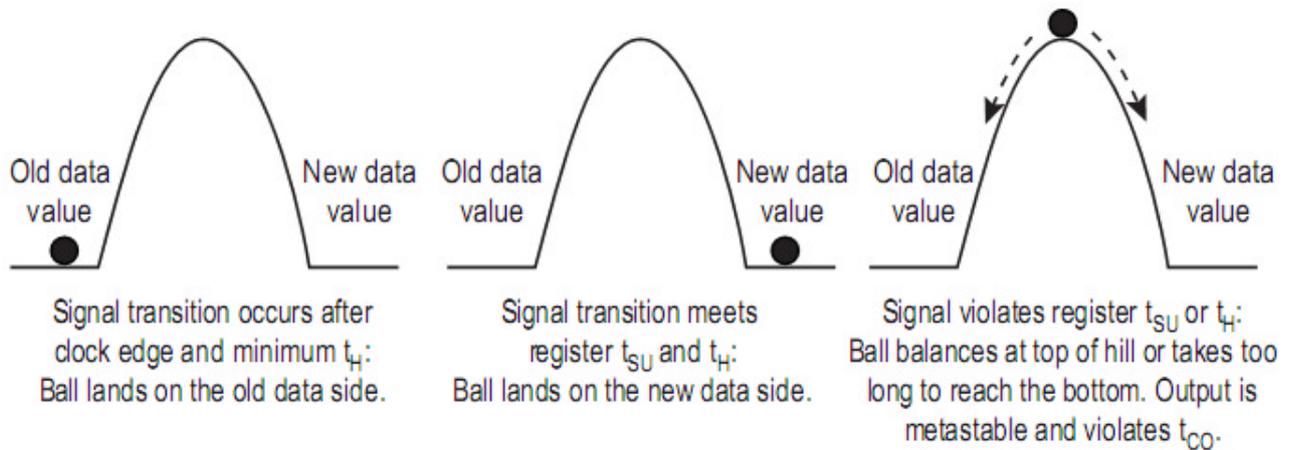
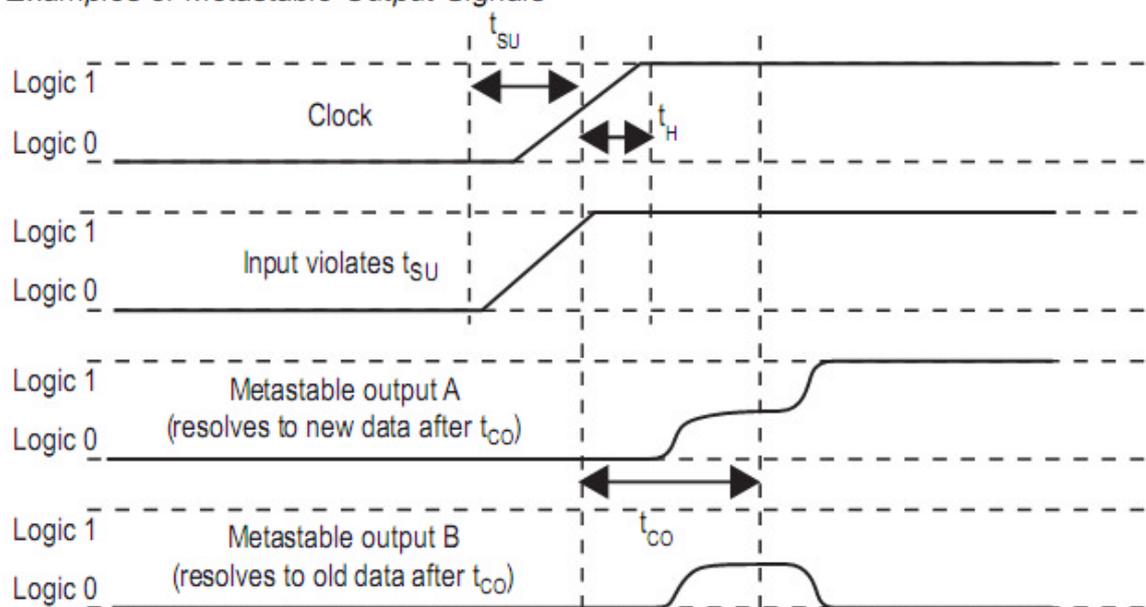


图 2 举例说明亚稳态信号，当时钟翻转时输入信号从低电平翻转到高电平，违反了建立时间的要求。数据输出信号从低电平开始进入亚稳态，然后在高低电平之间徘徊。信号输出 A 输出了输入数据的新逻辑 1，输出 B 回到了输入数据的原始逻辑 0 状态。在两种情况中，输出翻转到逻辑 1 或逻辑 0 的延时都超过了寄存器定义的  $T_{CO}$ 。

Figure 2. Examples of Metastable Output Signals



### 什么时候亚稳态导致设计失败

如果数据输出在下一个寄存器获取(capture)数据之前稳定在一个有效的状态，那么亚稳态信号不会消极地对系统运行产生影响。但是如果亚稳态信号在到达下一个寄存器之前达到 0 或 1 的稳定状态，这将导致系统失效。继续抛球的比喻，当球到达丘的底部花掉的时间超过指定的时间(寄存器  $T_{CO}$  时间加上寄存器路径上的时序裕量)，系统就会失效。当一个亚稳态信号在指定时间内没有判决(resolve, 意即稳定)，如果目标逻辑得到一个不一致的逻辑状态，逻辑失效就会发生，这就是说对同一个亚稳态信号不同的目标寄存器得到了不同的值。

### 同步寄存器

当一个信号在无关的电路或不同的时钟域之间传输，在新时钟域内使用这个信号之前必须进行同步。

在新时钟域里的第一个寄存器的作用就是一个同步寄存器。

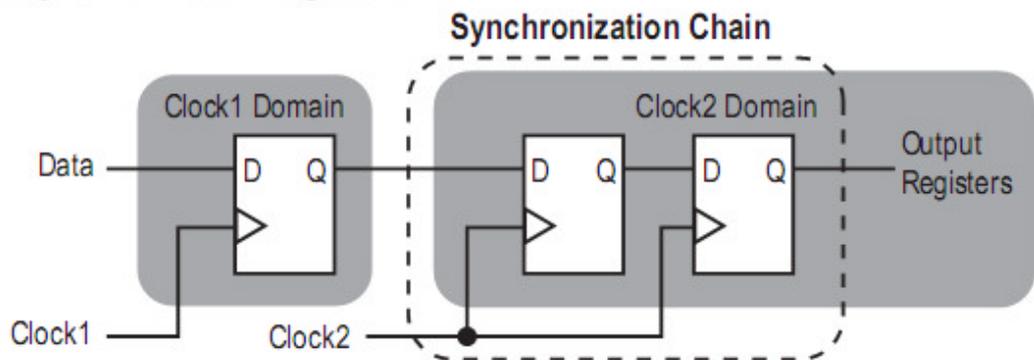
为了最小化由异步信号中的亚稳态造成的实现，电路设计者通常在目标时钟域内使用一系列寄存器(一个同步寄存器链或同步装置)来再同步(resynchronize)信号到新的时钟域内。这些寄存器为潜在的亚稳态信号到达整个设计的其它部分提供了一些时间使其到达一个稳定的值。同步装置中寄存器到寄存器路径上的时序裕量可以为亚稳态信号的提供稳定时间，就是亚稳态转稳态时间。

一个同步寄存器链或同步装置被定义成满足如下要求的一系列寄存器：

- 链中的寄存器是由同一个时钟或相位相关时钟驱动
- 链中的第一个寄存器是由无关的时钟域驱动的，或是异步的
- 除了链中最后一个寄存器，每个寄存器的扇出只有一个寄存器。

同步寄存器链的长度是同步时钟域中满足以上要求的寄存器的个数。图 3 中显示出了一个长度为 2 的简单同步链，信号的输出供给超过了 2 个目标寄存器。

Figure 3. Sample Synchronization Register Chain



注意相对于取样寄存器(capturing register)的时钟边沿，任何异步输入信号或在无关时钟域之间传输的信号可以在任何点翻转。因此在数据翻转之前设计者无法预知信号的翻转顺序或目标时钟的边沿。例如，一根总线上的异步信号(比如 16bit 的线)在一个时钟域中传输、同步，数据信号可以在时钟的不同边沿翻转，结果是从总线上收到的数据不可能正确。

设计者必须在电路中适应这种行为，比如双时钟 FIFO 逻辑存储数据或握手逻辑。FIFO 逻辑使用同步装置在两个时钟域之间传送控制信号，然后使用一个双口 RAM 对数据进行读写。Altera 为这个操作提供一个双时钟 FIFO 核，它包括不同级别的潜伏期并对控制信号进行亚稳态保护。另外，如果一个异步信号做为两个时钟域之间的握手逻辑的一部分，控制信号指明了什么时候可以在时钟域之间传输数据。在这个例子中，同步寄存器用来确保亚稳态不会干扰控制信号的接收，在亚稳态情况下数据在使用前有足够的时间达到稳定。在一个恰当地设计的系统中，设计的运行完全正确只要信号在使用之前就达到了一个稳定的值。

### 计算亚稳态平均无故障时间(MTBF)

失效之间的平均时间或者说平均无故障时间，取决于亚稳态提供的一个在各种实例中其导致的设计失效的平均时间估计。一个高的平均无故障时间(比如两次亚稳态导致失效的时间为几百几千年)表明一个设计是健壮的。平均无故障时间的要求取决于应用。例如，一个医疗急救设备对平均无故障时间的要求要高于一个消费型视频播放器。提高亚稳态导致的平均无故障时间减少了信号传输中在设备中导致亚稳态问题的风险。

亚稳态平均无故障时间(The metastability MTBF,即为由亚稳态决定的平均无故障时间)对一个特定的信号传输,或整个系统中的传输,可以由设计的一些信息和设备的特点来计算。一个同步链的平均无故障时间可以由以下公式计算:

$$MTBF = \frac{e^{t_{MET}/C_2}}{C_1 \cdot f_{CLK} \cdot f_{DATA}}$$

$C_1$  和  $C_2$  是常数,依赖于器件工艺和操作环境。

$f_{CLK}$  和  $f_{DATA}$  参数取决于设计规格:  $f_{CLK}$  是接收异步信号的时钟域的时钟频率,  $f_{DATA}$  是异步数据的反复频率(toggling frequency)。更快的时钟速率和更快的数据反复(faster-toggling data)会降低平均无故障时间。

$t_{MET}$  参数是亚稳态转稳定时间(vailable metastability settling time),或者说时序裕量大于寄存器  $T_{co}$  可以让潜在的亚稳态信号达到稳定值的时间。 $t_{MET}$  对同步链来说就是链中每个寄存器输出时序裕量的和。

整个设计的平均无故障时间由设计中的每个同步链决定。对一个同步装置来说失效率是  $1/MTBF$ ,则整个设计的失效率是把每个同步链的失效率加起来,如下:

$$failure\_rate_{design} = \frac{1}{MTBF_{design}} = \sum_{i=1}^{number\ of\ chains} \frac{1}{MTBF_i}$$

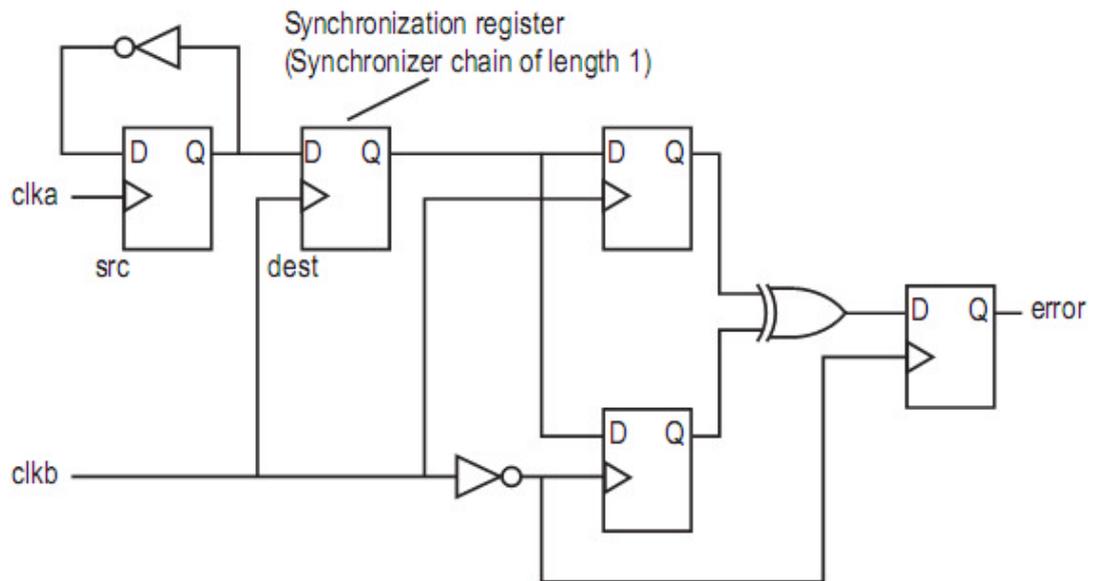
整个设计的亚稳态无故障时间是设计的失效率的倒数( $1/failure\_rate_{design}$ )。

设计者使用 Altera FPGA 不需要手动计算这些,因为 Altera 的 QuartusII 软件在其工具里已经包含了亚稳态参数。QuartusII 软件报告了检测到的同步链的 MTBF 并提供整个设计的亚稳态 MTBF。

### 亚稳态常识

FPGA 供应商决定 MTBF 计算公式中描述亚稳态的常数。描述亚稳态常数的难点在于对典型的 FPGA 设计平均无故障时间是数年,所以使用实际的设计在真实的运行条件下测量亚稳态事件之间的时间是不切实际的。为了描述器件的几个特殊的亚稳态参数,Altera 使用了一个有小的平均无故障时间的测试电路设计,如图 4 所示。

Figure 4. Test Circuit Structure for Metastability Characterization



在这个设计中，时钟 a 和时钟 b 是两个无关的时钟信号。到同步装置的数据输入每个时钟周期反复 (toggles) 一次(一个高的  $f_{DATA}$ )。同步链的长度为 1，因为同步寄存器输出到两个目标寄存器。目标寄存器分别在一个和半个时钟周期后获取同步寄存器的输出。如果信号在下一个时钟沿之前到达亚稳态，电路检测并取样到不同的信号，并输出错误信号。这个电路可以检测到很大部分的发生在半个时钟周期之内发生的亚稳态。

这个电路被复制到整个器件中来减少器件的影响，而且每个实例都被连续的测试以减少噪声的耦合。Altera 每个测试结构测量一分钟，并记录错误数。测试在不同的时钟频率下进行，MTBF 的  $t_{MET}$  结果被标在一个对数坐标上。 $C_2$  常数符合实验结果中曲线的趋向， $C_1$  常数与曲线成线性关系。

### 提高亚稳态的平均无故障时间

考虑到 MTBF 计算公式里的指数因子  $e^{MET/C_2}$ ， $t_{MET}/C_2$  对 MTBF 计算的影响较大。因此亚稳态平均无故障时间可以通过增强结构优化器件的  $C_2$  参数来提高，或者优化设计提高同步寄存器中的  $t_{MET}$ 。

### FPGA 的增强型结构

在 MTBF 计算公式里的亚稳态时间常数  $C_2$  和器件生产中的各种工艺因素有关，包括晶体管的速度和供电电压。更快的工艺和更快的晶体管可以使亚稳态信号更快的稳定下来。考虑到 FPGA 已经从 180nm 工艺提高到 90nm，晶体管的速度提升通常也提高了亚稳态的平均无故障时间。

但是，供电电压减小了，但门限电压并没有适当的减小。当一个寄存器进入亚稳态，它的电压大致是供电电压的一半。随着供电电压的减小，亚稳态电压非常接近电路的门限电压。当这些电压变的非常接近，电路的优势降低了并且寄存器需要更长的时间离开亚稳态。当 FPGA 进入 65nm 或者更低，供电电压达到 0.9V 或者更低，门限电压的考虑会比提高晶体管速度更重要。因此，亚稳态的平均无故障时间变得更差除非供应商设计 FPGA 电路去提高亚稳态的健壮性。

Altera 使用 FPGA 结构的亚稳态分析来优化电路提高亚稳态平均无故障时间。Altera 的 40-nm Stratix IV FPGA 上的结构提高和新发展的器件已经提高了亚稳态健壮性通过减低  $C_2$  参数。

### 设计优化

MTBF 计算公式里的指数因子告诉我们可以通过提高设计的  $t_{MET}$  指数级的提高同步装置的 MTBF。例如：如果给定器件和运行条件下的  $C_2$  是 50ps，则在  $t_{MET}$  上提高 200ps 则提高 MTBF 的  $e^4$  倍，即 50 多倍，然而如果提高 400ps 则累乘  $e^8$  是 3000 多倍。

另外，链的最坏 MTBF 对设计的 MTBF 影响很大。例如：假设两个不同的设计都有一个 10 个同步装置链，一个设计的 10 个链都有同样的 MTBF10000 年，而另一个设计的 9 个同步装置链有一百万年的 MTBF 但有一个的 MTBF 是 100 年。设计的失效率是每个链失效率的和。第一个设计的亚稳态失效率是  $10 \times 1/10000 = 0.001$ ，因此设计的 MTBF 是 1000 年。第二个设计失效率是  $9 \times 1/1,000,000 + 1/100 = 0.01009$ ，所以设计的 MTBF 约是 99 年，只比最坏情况的 MTBF 小一点。

换句话说，一个设计不好的同步装置决定了整个设计的亚稳态 MTBF。因为这个影响，对所有异步信号和时钟域的传输分析是很重要的。设计者或者设计工具供应商通过提高有最坏的 MTBF 的同步装置的  $t_{MET}$ ，对一个设计的 MTBF 有很重要的影响。

为了提高亚稳态的 MTBF，设计者可以通过在同步装置链增加另外的寄存器提高  $t_{MET}$ 。每个附加的寄存器到寄存器的时序裕量可以加到  $t_{MET}$  中。设计者通常使用两个寄存器来同步一个信号，但是 Altera 推荐使用标准的三个寄存器以有更好的亚稳态保护。但是，加一个寄存器就在同步逻辑中加了一个潜伏期，所以设计者必须衡量这样做是否可以接受。

如果一个设计使用 Altera 的 FIFO 核，有独立的读写时钟在交叉的时钟域，设计者可以提高亚稳态保护(同时提高了潜伏期)以得到更好的 MTBF。Altera 的 QuartusII 软件的 MegaWizard™ Plug-In Manager 提高一个可选的亚稳态保护选项有三级或更多的同步级。

QuartusII 软件同时业界领先的亚稳态分析和优化性能来提高  $t_{MET}$  在同步寄存器链上。当一个同步装置被检测出来后，软件会将同步寄存器的布局靠的很近来提高同步链中的输出时序裕量，然后报告亚稳态 MTBF。

★ 更多关于分析和提高 Altera FPGA 的 MTBF 信息参考 Quartus 手册第一卷中的” [Managing Metastability with the Quartus II Software](#)” 章节。

## 结论

当信号在无关的电路或异步时钟域中传输亚稳态就会发生。两次亚稳态失效的平均时间与器件的工艺，设计规格和同步逻辑中的时序裕量有关。FPGA 设计者可以使用一些设计技巧提高  $t_{MET}$  或增加同步寄存器中的时序裕量来提高系统可靠性和平均无故障时间。Altera 通过 MTBF 参数描述其 FPGA 并通过提高工艺来增加 MTBF。设计者使用 Altera FPGA 可以利用 QuartusII 软件的特点来报告他们设计中的亚稳态 MTBF，并优化设计布局以提高 MTBF。

## 更多信息

- 使用 QuartusII 软件处理亚稳态

[www.altera.com/literature/hb/qts/qts\\_qii51018.pdf](http://www.altera.com/literature/hb/qts/qts_qii51018.pdf)

- AN 473: 在异步时钟域中传输数据时使用双时钟 FIFO(DCFIFO)

[www.altera.com/literature/an/an473.pdf](http://www.altera.com/literature/an/an473.pdf)

PS:第一次翻译，有几处感觉不是很好。原文名：Understanding Metastability in FPGAs

译者：captainliuy@163.com