



小明和小强的故事

一直以来都在想写这么一个故事，这个故事的目的是为了说明清楚关于 timing 的一些概念。所以如果你对时序已经非常清楚的话，大可不必看下去以至于把自己反而搞糊涂了。

有一天天气很好，所以小明决定去小强家里。小明和小强家的门都是自动门，不是你想开，就可以开了的，而是在一定的时间后打开，而这时间间隔是固定的。为了让时间变得简单一点，我们假设它是一个小时。于是，小明在自己家门 4:00 打开的时候出去了。然后他就向小强家里走过去，小强家的门是 5:00 打开的，小明在这之前到达小强家门口。这样小明就顺利到小强家里了。

这个故事就讲完了。让我们重新把它推演一下，这个故事里面关键的是两点，一个就是小明到达小强家的时间，还有一个就是小强家的门的状况。这两个相对来说是两件独立事件。我们首先看一个小明从家里到小强家门口这么一个过程。首先是开门的时间点(Launch Edge)，开门需要一点时间，你不能在它刚开的时候就出去吧 (Tco)。最后就是从门口走到小强家门口的时间(Tdata)，这个时间是不一定的，因为有很多方式可以到小强家里。我们会比较关心极端事件，就是最快的时间，和最慢的时间。

这还没结束呢，最容易忽略的，其实就是时间的问题。好比小明以为他是四点钟出门的，但其实他家里的钟其实晚了 10(Tclk1)分钟的。如果我们以一个标准时间来判断小强和小明家的钟。而他们家里的钟只有可能比标准时间要晚一些 (时钟传递到门的延迟)。差不多可以了，我们看到这么一个公式

到达时间 = Launch Edge + Tclk1 + Tco + Tdata

如果非要给这些值赋一个固定的值，那么我来具体看一下。这里面 Launch Edge, Tco 都是固定的，我们假设他们为 Launch Edge = 4:00, Tco = 5。如果我们说小明花了半小时走过了中间的路，那么我们可以知道，当小明到小强家里的时候应该是 4:00 + 5 + 10 + 30 = 4:45。

好了，我们再来研究一下小强家的门。这个比较简单一点，就是小强家时间不准的时间，假如说为 5(Tclk2)分钟好了，再加上开门时间点 Latch edge。这其实说明门是晚开了 5 分钟。所以小强家开门的时间是：

开门时间 = 时间点 + Tclk2 + Latch Edge

这其实就是 5:05 了。

我们一直在说小明要在小强家门开之前到，那么究竟多久之前到他就可以安全的进去呢？这里面有一个建立时间 (Tsu) 好比要通过安检啦什么的。把这个值设为 3 好了，所以其实小明应该在 5:02 (5:05 - 3) 到是安全的。这点条件小明是符合了的 (4:46)。但是还有一个条件。小强看到小明会很开心，所以他要和小明聊几句才让他进去，我们管这个叫做保

持时间，在这段时间里面小明不能走了，走了的话，也就进不去门了。我们假设这个时间也是三分钟的话。就是说，从 5:02 到 5:08 之间，小明应该很安心，很稳定的呆在小强家门口他才能进去。小明进步了小强家可能不是什么大不了的事情，不过如果你的设计时序不合格，就要打屁股了。所以我们来看看什么情况下，小明进不了门。

首先我们俩安排几个参数。

Fmax: 什么是最理想的状况，就是小明走最长的路到达小强家门口的时候，恰好在小强家门口的建立时间点上。而我们把这中间的开门的时间间隔的频率称为 **Fmax**。以这个为例：小明 4:46 分到了，如果小强家的门在 4:49 分开就是最好的时间点了。而小强家的钟的延迟有给了我们 5 分钟的冗余。所以最短的时间应该是 44 分钟。这就是 **Fmax**。这也就是你的电路能够达到的最快的频率了。而我们规定的时间间隔大于 44 分钟就可以保证小明可以在建立时间点前到达。

我们给这个差异一个名字 **setup slack**。就是建立时间点减去小明的到达时间点。结果如果是正的，说明小明可以顺利到达，如果是负的，就不行。

刚才我们说过了，还有一个保持时间，我们来看看什么情况下保持时间会失败。我们给小明在找一个弟弟，这个弟弟呢会在下一次门开的时候出来，也来小强家。弟弟来的时候，小明就必须离开了。如果我们把小强家的钟再搞晚一点，晚了 20 分钟的话，那么开门的时间就会晚到 5:25。而保持时间点是 5:28 分。弟弟从 5:00 + 10 + 1 + 5 = 5:16 出发，假设他走了一条近路，很快的就到了，比如只花了 10 分钟，那么他在 5:26 分到了，小明就不得不走了。所以我们这里其实可以得到一个很有意思的结论，当发生保持时间问题的时候，往往是两种可能。

1. 相互之间的时间误差差距比较大
2. 中间的路径非常的短。

所以我们往往需要分析电路的最快表现来分析保持时间。

这里面也有一个 **hold time slack**。就是小明弟弟到的时间点减去保持时间点。

这个故事差不多讲完了，大家可能已经晕了，其实我自己也差不多了。但是我们还要再接再厉。如果把两个门开关的时间间隔变得不同。做时序分析的时候，工具会找到所有状况中最糟糕的那个状况来分析。不用担心听上去好像很复杂，这都是工具自己会做的事情。你知需要规定你想要的间隔时间，工具就会为了你的目标而努力。对这个故事的整个的了解让我们对电路的具体过程有更深入的理解。让我们来看看我们可以做些什么来规避 **Slack** 出现负值的情况。

首先看看 **setup slack**:

1. 把时间间隔变大，也就是降低频率（这是废话）
2. 把路径缩短，就是把最长的那条路径变短，使用各种优化手段。
3. 有的时候，我们会发现，其实小明根本不认识小强，所以我们完全没必要管他是不是能进小强家门。（**false path**）我们可以告诉工具，不要去管它。

4. 如果在 5:00 到不了, 6:00 到也可以, 有的时候我们可能并不在乎他什么时候到, 只要能到就好。我们可以设置为 **multi-cycle**。我们告诉工具说, 没关系不用太着急的, 下次开门的时候再到也可以的。

再看看 **hold time slack**:

1. 把时间误差变小。首先我们看看什么情况下可能导致小明家的钟差 5 分钟, 而小强家却差 20 分钟。如果这个时钟是一个 **gated clock**, 那么他的驱动能力是有线的, 所以时钟传递的延迟会因此而变大。我们可以使用 **PLL** 来驱动时钟, 这样的话驱动能力会加强, 时钟传递的延迟因此而变短。
2. 把最短路径变长, 这可以通过添加 **LCell** 来人为制造延迟。