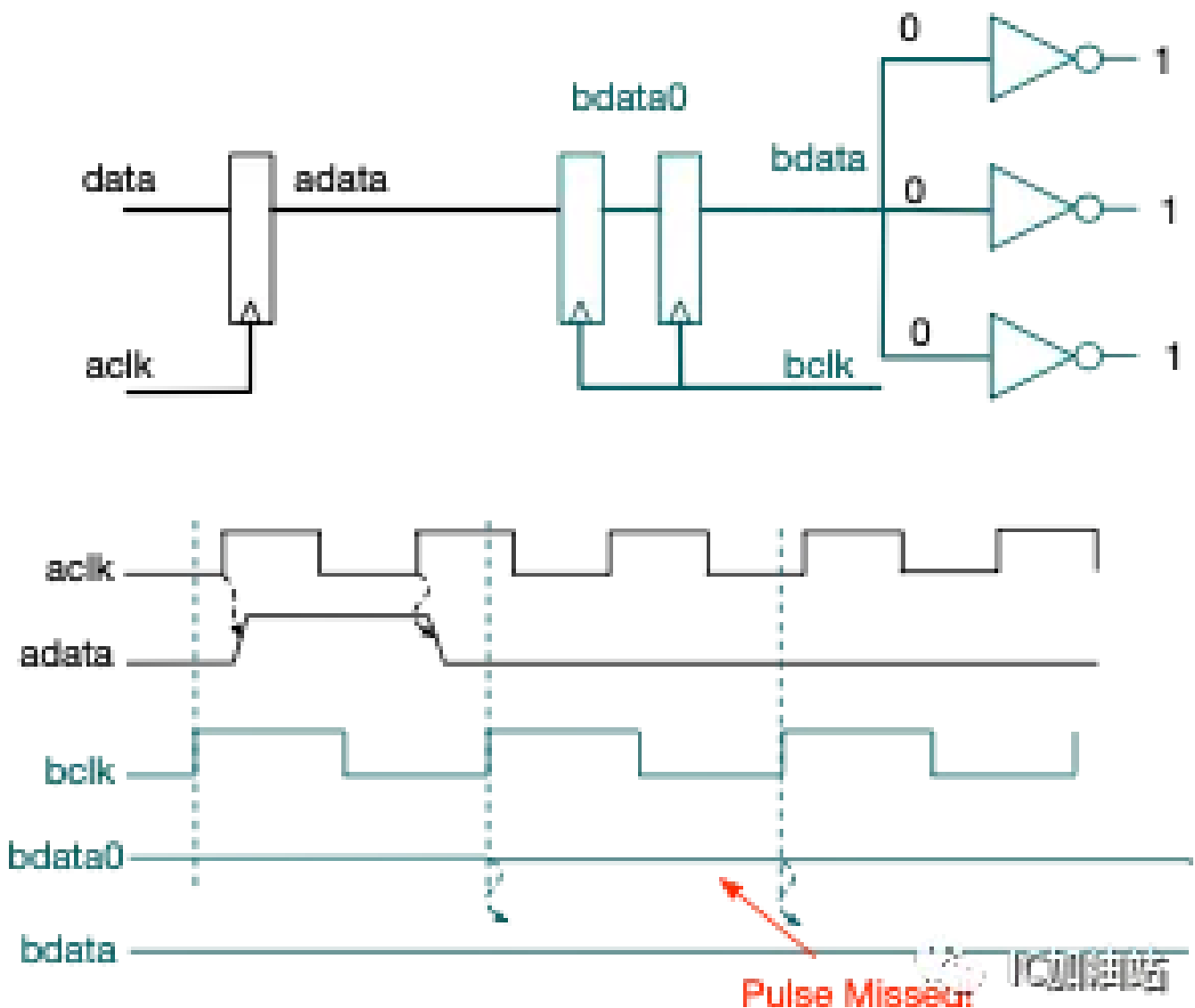


## 常见数电面试题Pulse Synchronizer -- CDC的那些事 (3)

原创 老李飞刀 IC加油站 5月26日

上一期老李挖了个坑，是关于同步一个时钟域的单周期脉冲(pulse)的问题。想当年这个问题老李在面试某几家大厂的时候被问到过不止一次，足以见得这是一个常考的知识点。在这篇里，老李带领大家破解这道常考面试题，让你在面试时能够游刃有余。

pulse信号在设计当中很常见，通常在某个时钟沿变高，在下一个时钟沿变低。我们上期举了下面的例子，可见当aclk频率比bclk频率高的时候，adata变高一个周期，那么有可能bclk的时钟沿根本看不到这个变化，或者有时候即使能被bclk采到一次，也可能无法满足3个沿的要求，导致无法用常见的2flop synchronizer来去sync。



那你肯定就会问了，那如果bclk的频率比aclk高，比1.5还倍高（3个沿的基本要求），比如10倍，一个aclk周期内可以见到bclk的10个上升沿，那不是肯定满足3个沿要求吗？难道这种情况也不能用2flop synchronizer吗？在回答这个问题之前，我们首先要回答一个问题：

## aclk时钟域的一个pulse，到了bclk时钟域内，应该是什么样的信号呢？

如果这个问题不好回答，那么换一个问法：

## 为什么要把aclk的pulse同步到bclk时钟域呢？

这才是触及灵魂的问题，对于普通的面试者，能够正确回答上pulse synchronizer当然不错，但是能够从根本上理解并且讲清楚pulse synchronizer存在的理由，那才是真正优秀的候选者。其实这个问题在工程师设计电路的时候也要自己问自己，当你无法给出确信的理你说你一定要把aclk的pulse同步到bclk时钟域，那你应该重新思考，也许你就会发现可能你真的不需要一个pulse synchronizer。

在电路设计中，经常我们需要设计一些pulse信号，比如说，有一个counter在不停地计数，每个周期加一或者减一，当counter的值等于一个特定的值的时候，我们就输出一个周期的pulse，可以用这个pulse来作为使能信号(enable)来做其他的事情，例如去set或者clear某个寄存器。反过来，也有可能用这个信号去作为某个counter增加1或者减去1的条件。再比如说，需要对memory进行读操作或者写操作，现在memory通常都有两个使能端口CE(Chip Enable)和WE(Write Enable)，CE和WE同时为1的周期表示要对memory进行写操作，CE为1但WE为0的表示要对memory进行读操作。再比如对于一个FIFO，push为高一个周期就表示给FIFO加入一个数，pop为高一个周期就是表示给FIFO减去一个数。以上这些例子，都说明了当这些信号为高时，就要有相应的操作发生，为高一个周期，就操作一次，再次为高时，就需要再操作一次，这是和另外一些状态信号(status signal)的差别。对于那些状态信号，它们为高或低只表示一种状态，而与它们为高为低经过了多少个时钟周期没有关系。我们在上一讲说到的用2flop来同步的单bit信号，几乎都是针对的那些状态信号。而对于active时需要进行相应操作的信号来说，很显然由于2flop synchronizer的限制，adata同步到bclk时钟域就无法保证持续相应的周期数，这里可能是最开始的例子bclk连一个cycle的pulse都没有，也可能是持续了多个cycle，自然不能用2flop synchronizer了。

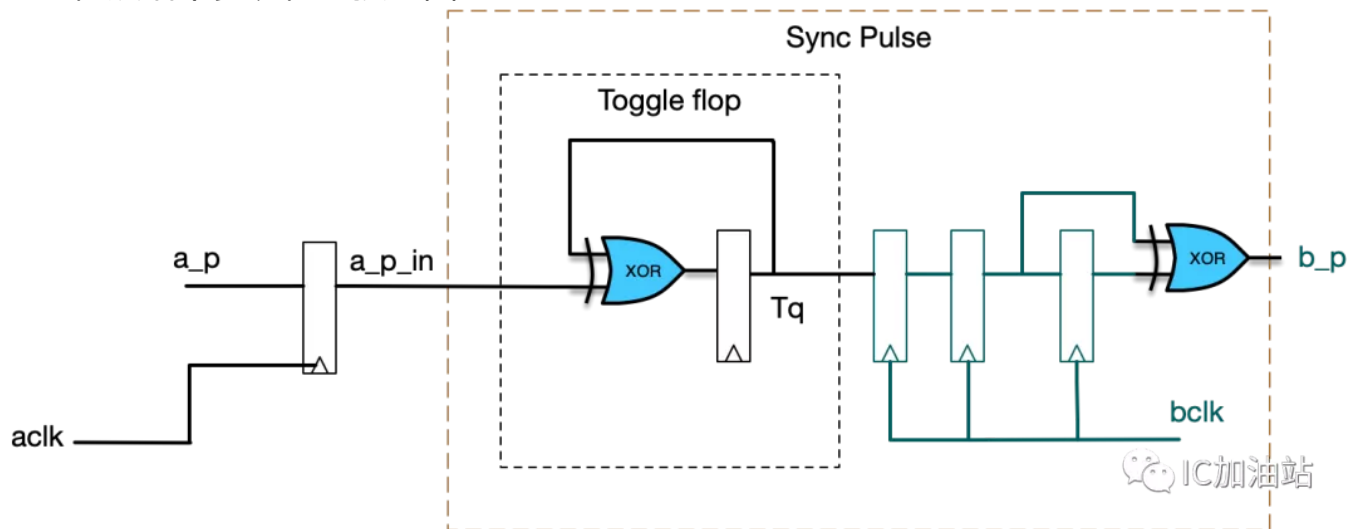
我们现在可以回答之前提出的问题了，当我们要同步aclk时钟域的一个单周期的pulse到bclk时钟域时，我们期望bdata是什么样呢？答案就是，**bclk时钟域也是单周期的一个pulse。**

那么如何克服2flop synchronizer的问题呢？咱们来看，由于pulse只持续一个周期，2flop synchronizer可能会miss掉pulse，那我们想个办法让产生过pulse这个之前发生过的事件记录在那里不就好了吗？这就是破解这个问题的思路：

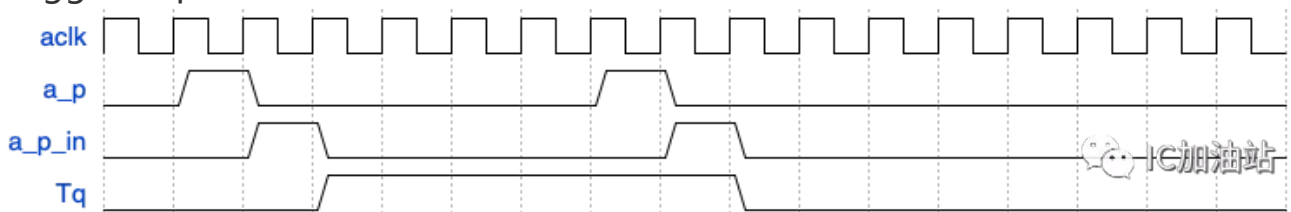
1. 将aclk时钟域的pulse信号转为一个level信号

2. 用2-flop synchronizer来同步这个level信号
3. 在bclk时钟域将同步过来的level信号转化为pulse

好了，废话不多说，直接上图



其中的关键就是图中所示的Toggle Flop。可以看到当输入是一个单周期的pulse时，里面Toggle flop的输出只会翻转一下。



这样我们就把一个pulse转化成为了level，这个level信号直到下一个pulse来之前都是稳定的，于是我们就可以利用2flop synchronizer来将这个level信号同步到bclk时钟域，然后我们再借助一个XOR和一个flop来重新创建一个pulse。

如果你面试的时候已经能够正确回答出上面的电路图，在面试官心里你已经及格了，但是距离完胜其他求职者还要再深入思考一些。

我们继续看，既然这个pulse synchronizer中间利用了2flop，那么2flop的3edge要求就必须满足，换句话说，我们转化成为的level的信号Tq要足够长。如果Tq不满足bclk的3edge要求，那么这个level信号我们就无法同步过去，也就无法产生bclk的pulse了。而Tq每次变化是由于aclk来了一个新的pulse，这也就是要求**aclk的连续两个pulse之间的间隔要足够大，要满足bclk的3edge要求。**

如果你回答出来了这个pulse synchronizer的局限性，那么面试官很可能会接着问，如果我不知道下一个pulse是什么时候来怎么办呢？老李当年还真被这么问到过，把老李差点整懵逼了。我们先来看，aclk时钟域最接近的两个pulse能靠多近呢，显然就是两个pulse中间只有一个aclk周期，这其实就是将aclk进行了2分频。那么相应的，Tq就是对aclk进行了4分频，每个Tq的level持续时间是2个aclk cycle，这2个cycle需要满足



于是，又给下一篇埋下坑了，下一篇老李会带大家了解CDC面试中最爱考察的asynchronous FIFO，敬请关注。

注：题图与内容无直接关系，我们这里讨论的pulse通常都是方波。题图就是吸引大家眼球用的。

如果你觉得这篇文章对你有所帮助，不妨点个右下角的“在看”，如果能够分享到群里或者朋友圈，老李就更有动力更新了。