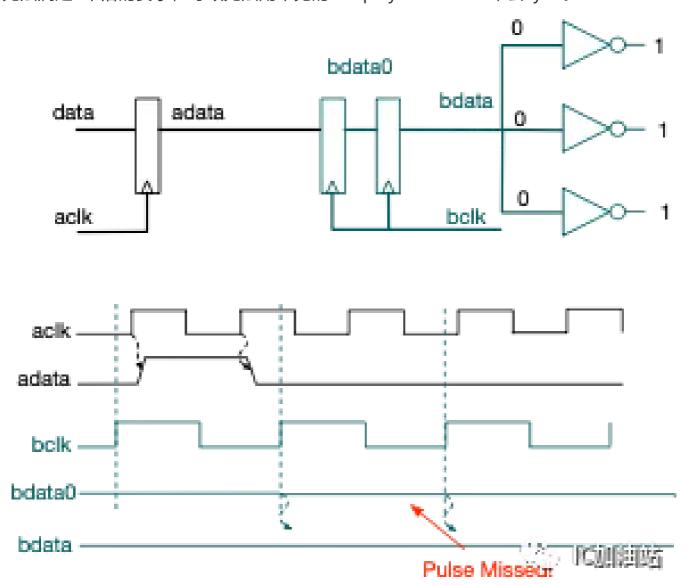
常见数电面试题Pulse Synchronizer -- CDC的那些事 (3)

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由说你一定要把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。

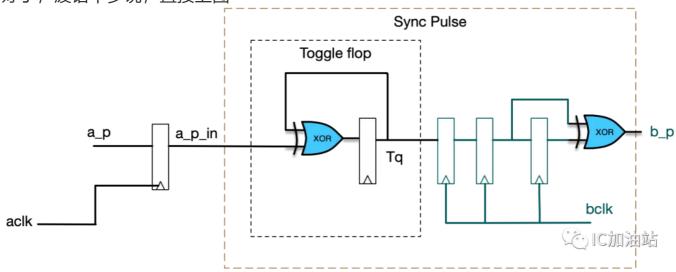
我们现在可以回答之前提出的问题了,当我们要同步aclk时钟域的一个单周期的pulse到bclk时钟域时,我们期望bdata是什么样呢?答案就是,**bclk时钟域也是单周期的一个pulse。**

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:

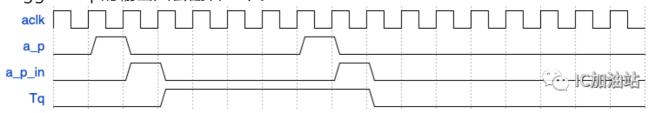
1. 将aclk时钟域的pulse信号转为一个level信号

- 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。



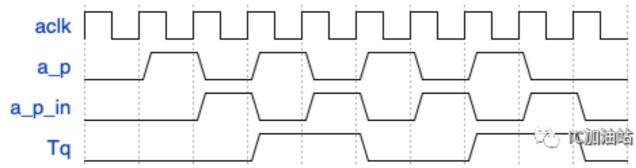
这样我们就把一个pulse转化成为了level,这个level信号直到下一个pulse来之前都是稳定的,于是我们就可以利用2flop synchronizer来将这个level信号同步到bclk时钟域,然后我们再借助一个XOR和一个flop来重新创造出一个pulse。

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。

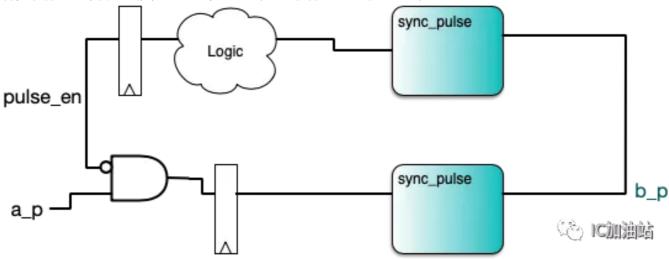
我们继续看,既然这个pulse synchronizer中间利用了2flop,那么2flop的3edge要求就必须要满足,换句话说,我们转化成为的level的信号Tq要足够长。如果Tq不满足bclk的3edge要求,那么这个level信号我们就无法同步过去,也就无法产生bclk的pulse了。而Tq每次变化是由于aclk来了一个新的pulse,这也就是要求aclk的连续两个pulse之间的间隔要足够大,要满足bclk的3edge要求。

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是什么时间来怎么办呢?老李当年还就真被这么问到过,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

bclk的3edge要求,如果满足,那么就可以保证bclk域也可以产生每个cycle的pulse的要求。



如果无法满足,那么我们就要另想办法,如果还是要使用pulse synchronizer这个电路,我们继续祭出反馈大法,这一方法我们上一篇已经见过。



本质思路就是我们不能让aclk域的pulse产生得很快,而是要等到pulse同步到bclk之后才能继续产生下一个pulse,于是我们要将b_p重新synchronize回到aclk域,作为放行下一个pulse的条件。

可是这个反馈的办法要求每个pulse产生都要同步回来,这样做的效率不高,有没有更快的办法呢?我们想象有一个细长的管子,管子的一头我们往里面塞玻璃球,每塞进去个玻璃球对应一个aclk时钟域的pulse,管子的另一头我们往外弹玻璃球,每弹出一个玻璃球对应一个bclk时钟域的pulse。(哎,是不是嗅到了一点FIFO的味道了?)那么除非假定管子无限长,那么我们往管子里塞入球的平均速率不能永远大于从管子里弹出球的速率,我们可以允许某一段时间塞入球的速率高,管子作为一个缓冲区,来容纳之前塞入但是还没有弹出的球,但是当管子塞满的时候,我们必须得停下塞入球的动作,等待弹出球的那边弹出球之后才能腾出空间来继续塞。也就是说,当管子不是无限长时,尽管两边的时钟速率不同,能够保持一一对应的条件是两边塞球弹球的平均速率是一样的。这就是利用FIFO来解决的思路。如果FIFO告诉aclk说FIFO满了,那么aclk域就得停止产生pulse,如果不满,就可以继续产生pulse,最后我们把FIFO里面的元素完全弹完,就可以做到一一对应了。

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。