

cdc_check

gray code, 多bit同步约束

需要保证gray code bit之间的skew(指的是到dest clk的data path长度), 少于1个source clk period, 最好控制在0.5 period以内。

到组合逻辑EN gate的路径约束 (cdc)

某些情况下, 是采用en的方式将source clk的data数据固定到确定状态, 也就是即使你过来的数据是X态不确定态也没有关系, 因为和EN与的话, 就是一个确定的0值 (OR 成固定值1也是一样的道理)。确定design确定到这样EN gate的data端的最大时间做为data路径的max delay约束, 也就是CDC PATH

到dest clk register D端的约束 (synccdc)

从source clk到dest clk的异步路径, 一般还是要约束一下异步PATH的长度, 一般是要小于< 一个dest clk period -- 属于通用约束

cdc buffer check

有的时候design设计者故意在这样的异步路径上放一些cdc buffer cell (可以就看着一个dont touch cell, 方便PR找到这个cell) 可以选择性的约束这段路径的max/min delay 主要还是为了方便找到这样的PATH 原理都是上面描述的3种check类型。

sync cdc cell

随着工艺的提升, foundry厂已经开发了集成式的打两拍的同步cell, PR不用再考虑后dest clk域的两级flop之间的D PATH长度, 工艺厂已经做好了, 直接用就行, 默认都是满足时序要求的 (类似于集成于的clock gating cell)

async fifo异步 data路径约束

在之前旧工艺情况下, 一般没有强调去约束异步FIFO内部DATA PATH

异步FIFO的原理前提是source clk指针同步到dest clk, 这个期间花的时间要比DATA连线直线到dest clock的时间要长, 这样才能保证采样到的是稳定的数据。

所以在先进工艺，比如40nm以下，最好约束一个异步FIFO data path时间（从source clk的data flop到dest clk data flop的时间要小于gray code同步的时间，一般是2个dest clock period。你可以用max delay的方式约束，也可以写个脚本，自动计算这个max delay 时间 $< 2 \text{ dest clk period}$ 。如果是同步打两拍的话，打三拍就需要 $< 3 \text{ dest clock period}$ 。

附：asyn fifo data 路径设计

一般来说有两种style。

接收读DATA采样分离式的FIFO data cell或在一起的方式

这种方式就需要设置如果没有读到指定ptr位置的fifo data的话，需要将它的output gate成0，此gate可以看作是上面提到的cdc check。可以使用dont touch专用类型的EN cell。用EN方式保证接收数据稳定。

接收读DATA直接采用data select选择的方式

这种方式是多数教材里讲的方式，直接选择数据，这种方式写RTL代码简单易懂。

以上两种方式其实都是一样的，只是style的不同。