CDC常见问题

- 跨时钟时未同步，直接使用。
- glitch引入，组合逻辑电路存在竞争
- re-convergence分别同步后，再一起组合逻辑使用 -- 分别同步可能有先后关系，导致逻辑行为错误，应该是使用vld+bus的方式同步多bit信号。
- async-reset sync 应该使用异步复位，同步释放的方式。

metastable亚稳态

它出现的原因是不同clock domain的data信号在未经过cross sync而直接使用，当采样到信号跳变沿时，会采样到不确定的状态（亚稳态），不确定状态在不同的fanout net上被当作的value值可能会不一样，可能会导致功能错误。

亚稳态的出现本身是一种概率，不好DEBUG且出现后的行为也是不确定，所以很难DEBUG

glitch

在同步之前不能是组合逻辑输出，否则可能会产生glitch在cross时出现问题，这也是为什么需要register out之后再cross的原因

多个信号分别同步之后，再一起作组合逻辑，也会出问题，因为分别cross时可能会出现先后到达的问题，造成接收错误，有以下几种：

1. CK1 to CK2 domain的多个信号分别同步之后作组合逻辑
2. CK1 to CK2 domain的多个信号分别同步之后没有直接作组合逻辑，但是后面又各自seq delay之后再作组合逻辑
3. CK1/CK3 to CK3 domain不同domain信号，分别同步到相同domain之后作组合逻辑

同一个信号，经过不同的同步器同步之后，也会出现类似convergence的问题。

常用cross跨时钟方式

- HANDSHAKE
- FIFO
- 控制信号跨时钟DATA信号直接过去，这里有两种
    1. 控制信号跨之后与DATA信号相与，控制信号未cross之前保持接收端register收到的数据是0
    2. 控制信号cross之后，作为接收端resiger update mux的sel信号，控制信号未cross时，接收端register数据保持不变。

async reset

异步复位需要采用异步复位同步释放的方式，并且产生异步复位需要由register out产生，不能用组合逻辑产生，否则可能会出现CDC问题。

参考文章

https://blog.csdn.net/a389085918/article/details/79963911