数字 IC/FPGA 设计 100 问

Our goal: help making good designs ...

---- A: 数字 IC/FPGA 设计问题

1. Async FIFO 的深度，必须是 2^N 吗？否则 gray code 无法绕回，并且每次只有 1bit 变化？

   不需要。async_fifo 只需要深度是 2*K 就行，gray code 就能正常工作。2^N 比 2*K 多于的部分，gray code 的头尾各去掉相同数目（(2^N - 2*K) /2）的计数值就行。

   如此减少后的 gray code，照样可以保证每次只有一个 bit 变化。

2. 请问：异步 fifo 内部的 ram 如果是自己用寄存器搭的，需要上电复位吗，用哪个时钟域的复位信号呢？ram 中的寄存器需要复位吗？

   一般不需要。

   有具体原因吗？为什么外部逻辑里的寄存器需要复位？外部数据流寄存器也不需要复位吗？

   控制逻辑需要，控制不复位会混乱。

   数字电路，一般分两条 path：data path，control path。

   基本原理是：control path 需要复位，data path 不需要。判断依据：只要电路仿真功能正常，初始状态的“X”是无所谓的。比如 APB 总线的 prdata 应该不需要复位，但是 pready 肯定需要复位。

   所以，不是所有得 DFF 都需要复位。

   那你可以进一步想想，给 datapath 不加复位，有啥好处？如果没有好处，那就干脆全部的 DFF 都加复位就好了。

   省面积吧？

   你得准确的分析为啥省面积了。把理由 1/2/3 的列出来。

   大概是减少了复位时钟树，DFF 中的逻辑也能减少。

   这不就 get 了吗？另：reset 走线也是要面积的。

   再进一步，走线要面积可以理解（自己看看 PCB 板子就能明白）。那如何确认 DFF 会省面积？

   到工艺库中看下这两个 cell 比较下。

   这不就结了吗。不是你不知道，而是你想不想弄明白。

   对于任何别人的结论，自己有疑问的地方，自己想办法弄清楚，确认好，慢慢你就有思想了。

3. 跨时钟域（CDC）处理，是 100%成功；还是有很小很小的可能（亿、十亿、百亿分之一）会错？

我认为如果电路结构正确，是 100%成功的。但是一般老师(教科书)讲：有很低很低的概率会错，然后这么低的概率可以忽略。
如果真的是这样，你可以想象：你家 CPU/GPU/手机 SOC 的主频多少 GHZ？里面有多少 CDC 的地方？1GHZ == 十亿次/s；你的 CPU/GPU/手机 SOC 每秒/天/年有死机吗？
sky 有个课详细讲解了跨时钟域设计的原理与方法：《数字 IC/数字电路/FPGA 设计_从入门到精通_合集》：https://ke.qq.com/course/3133628?tuin=64ce5e2a 的第 13/14 讲。

4. 单 bit 信号做跨时钟域（CDC）同步，是寄存 2 拍吗？怎么又有寄存 3 拍的说法。
如果是低速电路：寄存 2 拍可以了；高速电路：可能需要寄存 3 拍。

这儿，什么是低速/高速电路？
没有定论，根据制程（0.18um/0.13um/90nm/28nm）不同而不同。对于 90nm 来说，可能 400MHZ~600MHZ 是分界线。
在 sky 的 QQ 群里，有过一次单 bit 跨时钟域处理的讨论，讨论内容整理如下：
https://zhuanlan.zhihu.com/p/354550783。

5. async fifo 的 full/empty 标识准确吗？比如 full==1 时，async fifo 肯定满吗？
不准确。这个由 async fifo 的结构决定的。
a) full 做在 wclk domain；empty 做在 rclkdomian；
b) rptr 从 rclkdomian 传递到(sync 到)wclk domain 需要花费几个 wclk 的周期；
c) 在传递的这几个 wclk 周期中，full 信号的标识是不准确的；
empty 同理分析。
这个现象，在计算 async fifo depth 时，需要考虑。

6. 寄存器配置后过一段时间再使用，不需要做同步吧，中间没有逻辑。
是否 false path，自己分析下，肯定不是必须要同步处理。
如果不同步处理，你是怎么保证：寄存器配置后，再用的。保证刚配置的 1ns，Q 端有翻转，电路不会错。
这是风险点。

你这个问题，可以这么想：所有的 CDC 逻辑，都必须做寄存吗？
比如：guard 信号保护的 data？具体的例子：比如 async fifo，里面的 register file 的 dout。

风险清楚，你确信没问题，我就信你。
PS：即使 post sim，也许不能 cover 跨时钟域的所有状态。
1：reg 的每个 bit 不同的翻转。32bit 寄存器，都多少种可能；
2：2 个 clock 间，各种 phase 差的可能；
3：各种 PVTcorner 导致的 cell delay 差异；
所以，跨时钟域电路，功能的正确性是从设计的电路结构来保证的。

7. post-gate sim 时，跨时钟域的第一级 DFF 总要 setup/hold 报错，"X"一直往后传递，怎么处理？
这个是预期的正常现象（跨时钟域必然 violation）。处理方法：

a) 修改 SDF 文件。把所有跨时钟域的第一级 DFF 的 setup/hold 改为 0（如果 0 不行，可以尝试负值）。

b) 使用 sim tool 提供的选项。比如 VCS：

vcs +optconfigfile+async_dff.list

文件 async_dff.list 的内容如下：

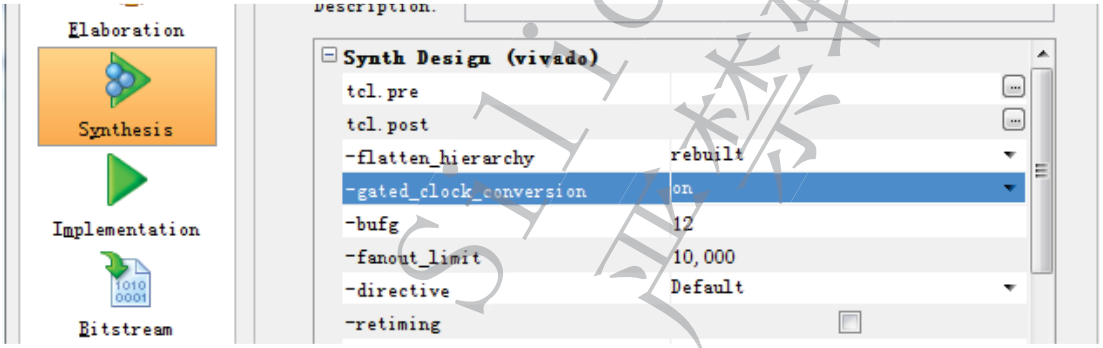
instance {tb_top.U_TOP.U_CORE.U_SYNC0_REG} {noTiming};

更推荐使用方法 b。

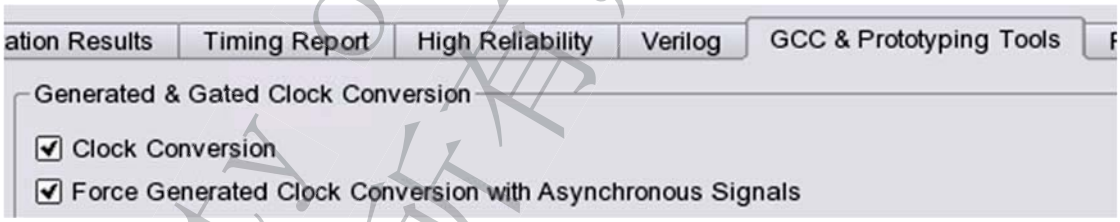
8. 请问：IC 中的 clock gating，在 FPGA prototyping 时，怎么处理？

原因：FPGA 中 clock tree(BUFG/BUFR)是有限的，但是一般 IC 中的 clock gating 非常多，每个 clock gating 都会占用一根 tree，数量不够怎么办？

解决：在 FPGA 中，把 IC 的 clock gating 全部变成 data enable。在 vivado 中如下操作：



在 synplify 中，如下操作：



9. 请问：为什么能把 clock gating 变 data enable？

想想 clock gating 是怎么来的。比如 DC 综合时，怎么自动插入 clock gating cell 的。clock gating 具体原理，看 sky 的文章《数字 IC/FPGA 设计基础_门控时钟》：https://zhuanlan.zhihu.com/p/354543957 。

10. clock 在 design 中，在组合逻辑/时序逻辑中穿来穿去，有啥不好？

clk 就是清纯小少女，得细心对待，不要乱搞。

时钟就是时钟，怎么能带载（驱动一般的组合逻辑），用 clk 驱动 celld 的非 clkpin 呢。

时钟信号从时钟树里拉了出来，这种情况下往往产生的输出还会继续用作时钟（比如:clock gating，clk divider 等）。绝大多数（99.999999%）情况，是这样的。

clk 带负载（drive）：CK Pin of a cell；包括：DFF/LATCH/SRAM/CLK gating 等的 CK 端。clk gating 是可能 drive 组合逻辑的地方。以上是 99.999999%的情况。
clk 在组合逻辑/时序逻辑中乱串，导致 STA 时序分析时，timing 容易 violation，特别是 hold。因为“乱串”，导致 clk 的 skew 变大。
另外，clk 得干净（没有 glitch），在组合逻辑“乱串”，肯定有 glitch。

11. IIC 的 SCL PIN，在芯片内部，可以直接当 CLK 用吗？

IIC 的 SCL，一般不要拿来直接当 clk 用。SCL 在系统（PCB 板级）级可能是多 device 驱动的，从 PCB 进入，可能会有毛刺。
具体看看 ARM/Synopsys 的 IIC IP，别人是这么玩的吗？为啥不呢？
做之前，参考下大佬的，没有坏处。

12. 芯片再次 tapout，代码中只改变了一点点逻辑（比如删除个 IF，case 的语句中改变一点点输出），后面的所有验证都要重新跑吗？

如下，都得再重新做一次：
1：所有的 testcase 跑 sim；
2：跑 formality/lec 对比；
3：postgatesim；
4：STAsignoff；
做数字 IP/SOC 设计，需要细致，认真。不要有侥幸心理，侥幸不测试，基本就是 bug。现在 tapout 一次太贵（比如 28nm，MPW 光给 foundry 生产费用就在 50 万左右），光生产就得 2～3 个月时间。如果有 Bug，再来一次，至少 6 个月没了。对公司来说，损失很大。

13. UART 可以自适应 baud rate 吗？

Howtowork（needahigherfreqclk）：
1：Training phase：
Afterpoweronorreset,firstsendseveralbytesofsynchronous data pattern,thenlockthebaud rate;这个 sync data pattern 是大家默认好的已知 pattern。
2：data transfer phase：
usethelockedbaudratetododatatransfer；
怎样设计 training phase 的 data pattern，这个自己想了吧。具体，可以参见 ST 的 STM32 MCU 的 datasheet，UART 部分。
UBS 协议，每个 package，也会传送几个 bit 的 freq sync 同步头；来修正 data rate；DDRPHYDQtraining，也是类似的做法。这个的目的：修正各个 DQ bit 线上的 delay 的差异。

14. CNN HW 架构参考资料？

1：Nvidia 开源的一个 CNN accelerator，有源码有设计文档。具体见 github：https://github.com/nvdla。
2：斯坦福课程：http://vision.stanford.edu/teaching/cs231n/

15. CNN 加速器，只有 HW 工程师，能开发吗？

难。因为需要 HW 跟算法合作的。如果没人给你提供网络 model，做好网络的压缩与量化，很难做出 HW 架构的。即使做出来，也没有网络可以跑 demo。
另外，CNN HW 通常需要有自定义的“指令”的，需要用 compiler 把某个做好的网络结构（网络层次关系，每层的运算，每层的 weight）compile 到 HW 架构定义的指令，所以还需要设计一个专用的 compiler。

16. 旁边的人和我都是说写 verilog 时，大脑里要有相关的数字电路，但是我自己一直想不出来。

这个概念是对的，Think in HW。
因为你没有 HW 的思想（不了解数字电路基础，底层逻辑单元结构与功能(AND/OR/NOR/XOR/MUX/DFF/Latch)，加法器/减法器/乘法器结构，verilog code 与底层逻辑单元的对应关系，...），需要恶补数字电路的基础知识。
可以看看这本书：《CMOS VLSI Design A Circuits and Systems Perspective》。
如果自学费时费力，也可以听听 sky 的课：《数字 IC/数字电路/FPGA 设计_从入门到精通_合集》：https://ke.qq.com/course/3133628?tuin=64ce5e2a 。

17. 17. 什么是 hw 架构呀？

HW = hardware.
架构：对于模拟电路，就是 PLL 的基础结构，有几个大模块；AD/DA 的基本方法，比如 SAR 结构的 AD。
对数字电路，比如做一个神经网络加速器，怎么划分功能模块，怎么跟 SW 交互，怎么访问 DDR 里面的数据，怎么组织运算，内部 SRAM 的数据怎么摆放，数学计算怎么切分 pipeline，都属于架构范围。
举个例子：修高楼，那个设计图纸就是架构。

18. 能否设计一个 DFF cell，让其：setup time 为负值，比如：-0.02ns？hold time 为负值？setup time + hold time 为负值？

某个 DFF cell,setup time/hold time 可以分别为负值；但是 setup time + hold time 肯定是>0 的正数。具体见免费课的分析讲解：
https://ke.qq.com/course/379407?taid=3087394291436047&tuin=64ce5e2a

19. 服务器 CPU 核很多，VCS 仿真能用多核吗？

这个不清楚。可以同时多跑几个 case，建多个目录。

20. modelsimstartsimulation 的时候，如果 enableoptimization 信号被优化掉了，如果不使能这个选项就会报错。

rtl compile 的时候（vlog）就强制不要 opt。
不开 optimize 的 run.do:

```
vlog -00 -vlog01compat +define+ SIM_DEBUG_MODE -f flist.f -f flist_trac.f
vsim -c +nowarnTSCALE -L ./work -novopt -l load.log tb
```

开 optimize 的 run.do:

```
vlog -04 -vlog01compat +define+ SIM_DEBUG_MODE -f flist.f -f flist_trac.f
```

```
vsim -c +nowarnTSCALE -L ./work -l load.log tb
```

————————————————————————————————————

顺带打个广告，我司数字 IC/FPGA 设计培训课程（**腾讯课堂**）：

a) 《数字 IC/FPGA 设计入门_合集》：
https://ke.qq.com/course/3133628?tuin=64ce5e2a

b) 《PPGA 设计入门》： https://ke.qq.com/course/3067626?tuin=64ce5e2a

c) 《On-Chip-Bus 精讲》： https://ke.qq.com/course/2900266?tuin=64ce5e2a

d) 《数字 IP 设计实例_A》：https://ke.qq.com/course/3132227?tuin=64ce5e2a

e) 《数字 IP 设计实例_B》：https://ke.qq.com/course/3200590?tuin=64ce5e2a

f) 《数字 IP_FPGA 设计实战》：https://ke.qq.com/course/3292002?tuin=64ce5e2a

————————————————————————————————————

作者介绍（QQ 技术交流群：877205676）：

sky：2006 年电子科大毕业；前 Verisilicon Senior Staff Engineer；数字电路前端设计从业 14 年；主要做视频 IP 设计（H.264/H.265 编解码器设计，JPEG 编解码器设计），CNN 加速器 IP 设计。参与 7 颗 ASIC/SOC 芯片设计（量产 3 颗）。目前申请 3 篇国家发明专利。

公司主页：http://www.siliconthink.cn