

数字 IC/FPGA 设计 100 问

Our goal: help making good designs ...

---- A: 数字 IC/FPGA 设计问题

21. 想请问一下，两个 256bit 的有符号数相加有什么优化方法吗？FPGA 上，250MHZ 时序不过。

如果用 ip 的话，可以选 pipeline 级数的嘛。DC 可以调用 DW，有 pipeline 的加法器；fpga 不知道是否有。

如果没有，可以考虑：

a): 自己拆分把。2T 做完(或更多 T)。

第一 T: 低 128bit 相加；

第二 T: 低 128bit 的 cin + 高 128bit 相加。

b): 曲线救国：

1): 使用 DesignWare 的 pipeline 的加法器；

2): synplify 综合；

3): fpga tool 读入该部分的 EDIF 网表；

调用 DW 的基础 IP 库，比我说的手工拆分 2~4T 做的面积更小。人家是从加法器结构出发，在底层加入 pipelien 的。

PS: 乘法器/除法器基础功能块，也类似操作。

22. ASIC 设计调用了 DesignWare 的 IP，在 fpga emulation 时，怎么弄？

Method A:

a) 找到跟 DC 相同版本或更高版本的 synplify_pro 综合工具；

b) 用 synplify 综合这段逻辑；综合时把 linux DC 下的 “dw_verilog.v” 文件加入到 synplify_pro 综合工程中；

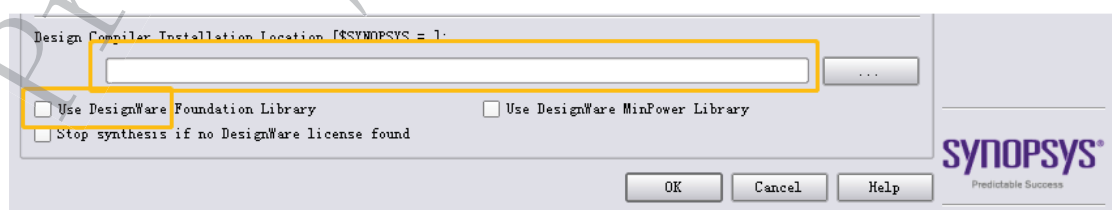
c) 产生网表(netlist)，在把网标读入 FPGA 工具(ise/vivado/quartus)；

注意：stage b)的具体操作，百度。

Method B:

a) 找到跟 DC 相同版本或更高版本的 synplify_Premier 综合工具；

b) 在综合工程的 "Implementation Options" 中，找到 "Verilog" 页，再做如下设置(传说 Design Compiler Installation Location 可以随便给个 path，sky 不确定)：



- c) 用 synplify_Premier 综合这段逻辑;
- d) 产生网表(netlist), 在把网标读入 FPGA 工具(ise/vivado/quartus);

23. xilinx 的 FPGA, 在 implement 后, 对设计的顶层 IO 总会加 IBUF/OBUF, 有啥用?
这个请先看看一个真实芯片的结构。FPGA 也是 ASIC 芯片, 那些引脚也是通过 PAD, 金属线(wire bounding)连接的, 所以 IBUF/OBUF 基本就是 PAD+wire bounding 的“替身”。
能不加吗?

24. 请问您知道一片 ddr 如何被多个用户接口控制吗?

通过 AXI 总线是可以的。

DDR Controller (AXI slave 接口) -> AXI bus interconnector --> AXI master (很多个)。

PS: 在 xilinx 中, DDR Controller 是 MIG。

25. xilinx 的 MIG 核(AXI 接口), 跑起来, 为啥实际带宽只有理论带宽的 20% 不到? 太低了。
检查如下东西: 你 MIG 核的 AXI 接口, data width 多宽? 你的 user IP 得 AXI master 口, data width 多宽?

中间有挂 AXI data width convert。

那转换后, user IP 的 AXI burst length 转换到 MIG 口处, burst length 还有多长?

多谢。还有别的考虑吗?

你 user IP 支持 AXI read cmd out of standing 吗?

啥是 AXI read cmd out of standing?

这样啊! ? 具体见腾讯课堂:《On-Chip-Bus 精讲》,

<https://ke.qq.com/course/2900266?tuin=64ce5e2a>

26. Xilinx zynq 器件 (比如 zynq7020) PS 端的 ARM CPU, 怎么跟 PL 的 user IP 端交互数据?

这个跟一个普通的 SOC 中, CPU 怎么跟 user IP 交互式一样的。

大量数据的交互:

- a) 器件留有 4 个 HP AXI master 接口 + 2 个 GP AXI master 接口;
- b) user IP 可以通过这些接口访问 PS 端的 DDR;
- c) ARM CPU 也可以访问 PS 端的 DDR;
- d) 所以, 就可以靠 PS 端的 DDR 来沟通数据了;

注意: CPU 的 cache 可能是 enable 的, 怎么保证 CPU/user IP 看到的 DDR 里面的数据一致性, 是个需要注意的地方。

zynq 在 PS 端, 还提供片内 SRAM 空间, 同样可以跟 CPU 交互数据。

CPU 访问 user IP 的寄存器:

- a) 器件留有 2 个 GP AXI slave 接口;
- b) CPU 可以通过这 2 个口访问 PL 端的 use IP;
- c) vivado 提供多个协议转换的 IP: axi<-->axi lite, axi lite <--> APB;
- d) 然后 CPU 就可以访问 user IP 的寄存器了;

27. 请问：zynq 怎么开发使用？

zynq 开发，在公司里，一般是 fpga 工程+driver 工程师一起配合设计开发的。
一个做数字电路设计，建 vivado 工程；一个专门写 driver，系统应用与协助 debug。

28. xilinx 的 ILA 抓信号(比如 DFF 的 Q 端输出的 result[7:0])，在图像界面看到的波形，一个周期代表什么意思？是原始信号的一个周期吗？

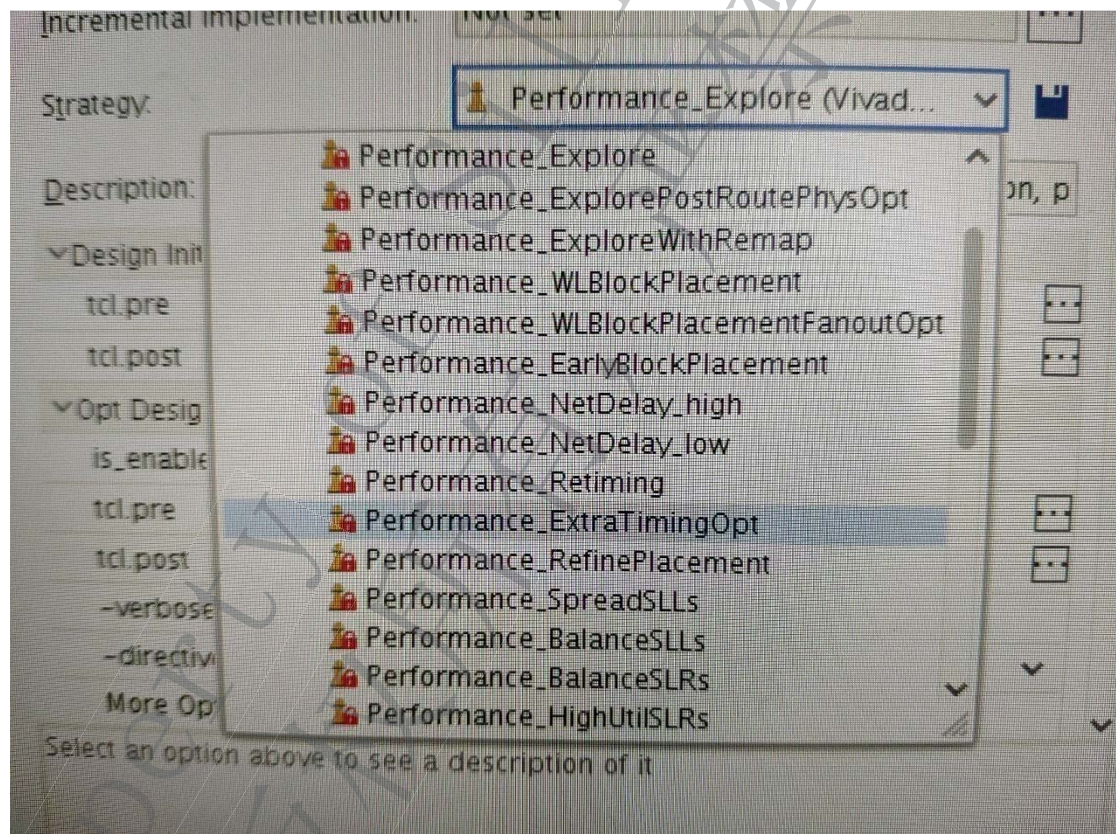
不是。图像界面看到的波形的 1 个周期，是采用时钟每个周期看到的值。假设：result[7:0] 用 clk0 drive:

- a) 如果采用时钟就是 clk0，则看到的波形就是 clk0 每个时钟周期看到的值，跟仿真看到的类似；
- b) 如果采用时钟就是 clkx，则看到的波形就是 clkx 每个时钟周期看到的值，跟仿真波形可能不一致；

如果让你设计一个 ILA 的数字 IP，你咋做？想想，就明白了。

29. 请问：vivado 布线后，有 0.3ns setup violation，怎么弄？

不想改代码，综合，布局布线时，使用 timing 优先的策略。vivado 叫：strategy。



请问哪个是 timing 优先呀？

既然找到了，看 vivado userguide 啦。

PS: 如果不行或是 violation 太大 (0.5ns 以上)，就只有该 RTL code，切分 pipeline 或是修改电路结构了。

30. 在仿真时，怎么把 fpga 器件的仿真库添加进入？比如：unisims, xcelib 等。

- a) Modelsim 的话，可以按网上说的，先把这些库编译了；

- b) 更通用的做法，用 verilog sim tool(Modelsim 也可以的)提供的一般用法: -y 。

```

1 +define+H264_RTL_SIM
2 +incdir+../rtl_src/top
3
4 -y /home/yuxiang/proj/sim_libs/unisims
5 +libext+.v
6 -v /home/yuxiang/proj/tsmc_lib/tsmc13.v
7
8 ../pattern/h264_lite_tb.v
9 ../pattern/trc/h264_sim_cur_ref_ini_dump.v
10 ../pattern/trc/h264_str_stream_trc.v
11 ../pattern/trc/h264_cycle_bitrate_monitor.v
12

```

31. 如何提高 modelsim 的仿真速度?

Normal:

```
vlog -O0 -vlog01compat -f flist.f
```

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

提高速度:

```
vlog -O4 -vlog01compat -f flist.f
```

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

32. 对于大型数字 IP/SOC 开发，SWtool sim 速度太慢，有提高效率的办法吗?

- a) 上 FPGA。联合 driver 同步开发，跑 fpga emulation。

- b) 土豪，买硬件仿真加速器。比如:

Mentor (Veloce2): Veloce2

https://www.mentor.com/products/fv/emulation-systems/veloce?sfm=free_for
rm Emulator – Mentor Graphics

Cadence (Palladium): Palladium Z1

https://www.cadence.com/en_US/home/tools/system-design-and-verification/acceleration-and-emulation/palladium-z1.html
rise Emulation Platform
(cadence.com)

PS:Synopsys 也有类似的东西。方法 a)/b) 有啥优劣，think it。

关于仿真实验，可以看看 sky 这篇文章《数字 IC/FPGA 设计基础_仿真实验方法》:

<https://zhuanlan.zhihu.com/p/354570641>

33. 公司有代码是 VHDL，看不懂，怎么办?

verilog/vhdl 是可以相互调用(instance)的。EDA tool 也支持 verilog/vhdl 在一个 project 同时读入进行处理(Debussy/Verdi/VCS/DC/Modelsim/Vivado)。

但是 VHDL 代码看不懂，怎么搞?

有 tool 可以把 VHDL<---->Verilog 互转的。

用过，转出来的代码很“可爱”，害怕转换后 function 不对，怎么办?

用 formal 对比两个版本。EDA Tool 比如: formality, LEC。

34. vcs 仿真 hold 为负数时 DFF 的出错。

- 试一下 RTL 代码编译时，命令行加上 `+define+NTC`（改变 cell library 的 verilogmodel 的行为，允许 negative hold time）；
- 看看仿真 tool，如下选项打开没有：`-negdelay, +neg_tchk`；以上是 vcs 的，不同的 tool，可能略有不同；

35. 如何设计奇数分频的时钟分频电路？

- 同时用 `negedge/posedge`，再 `and` 起来。可以做到 50% duty cycle。
- 如果不要求 50% duty cycle，可以按“每多少个周期输出一个时钟周期”的方式处理。使用 clock gating 的方式处理，只用 `posedge` 的寄存器就能做到。

注意：方法 b)，可以实现小于 1 的任意分频，含小数分频。比如 8/9 分频。

分频后的 clk 要干净（没有 glitch），能用 DFF 的 Q 端 or clock gating cell 直接输出的，就不要用 combination logic 输出。

36. 既然两个寄存器间的时钟偏移可以提升两个寄存器的最大时钟频率，那么如果我把两个寄存器间的时钟偏移设置非常大的化，岂不是可以使整个系统的时钟频率跑到非常高，接近于 1/Thold，但总觉的这样又不对。求大神给我的误区指点。

时钟偏移 == clk skew （尽量用英文，免得歧义）。

你这 2 个 DFF 是爽了，这 2 个 DFF 的邻居呢（前后级 DFF）？

37. 如果我两个 reg 是多周期的话，同样可以把频率提高吗？有什么弊端呢？

你通过设置 `multi_cycle_path` 是把 clock 的频率是提高了，但是 data throughput 呢？你得多个周期，数据才能更新一次啊。

$\text{Process performance} = \text{clk_freq} * \text{data_throughput}$ 。

38. 如何提高一个数学计算单元的计算效能（数据吞吐率）？比如：CNN，矩阵乘法。

不知啊，可以从如下方面分析：

运算量在那儿摆着，你有啥选择：

- 优化算法，减少计算量；
- 摆更多的运算单元（乘法器，加法器等）；
- 提高每个乘法器的利用率（争取每个周期都在做有效计算）；
- 提高时钟频率；

39. 请问：DC/STA 有哪些好的入门资料？

DC 综合：《AdvancedASICChipSynthesis》2ndED；

STA：《Static Timing Analysis for Nanometer Designs——A Practical Approach》；

下载方式：Baidu，或者 QQ 技术交流进群（877205676），共享文件找。

40. 请问：有没有前辈解释一下这个 casex 语句怎么用？

看官：正式代码就不应该出现 casex 这种语法。

有可能用到 casex/casez，当表示 one-hot 译码器时。还有别的 coding 方法吗？

看官：直接把逻辑写出来。

请给个例子，就实现 4 bit one-hot 译码功能。

看官：假设 di 是 4bit。

```
assign y = ({4{x[0]}} & d0) |  
           ({4{x[1]}} & d1) |  
           ({4{x[2]}} & d2) |  
           ({4{x[3]}} & d3);
```

牛逼🐮。真是这么回事。

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>