

鲁棒性检查 (Robust Verification)

鲁棒性检查，也就是将电路时序检查的条件合理地变严苛

On-Chip Variations

1. 建立时间检查 (Setup Time Check)

通常来说，工艺和环境参数在芯片的不同部分可能不一致

由于工艺差异，芯片不同部分中的相同 MOS 晶体管可能不具有相似的特性。这些差异是由于模具内的工艺变化造成的

差异的主要成因包括：

- 沿芯片区域的IR压降（电阻压降）影响局部电源供应
- PMOS或NMOS器件的电压临界点变化
- PMOS或NMOS器件的通道长度变化
- 局部过热点造成的温度变化
- 互连线金属蚀刻或厚度变化影响互联电阻或电容

These differences can arise due to many factors, including:

- ❑ i. IR drop variation along the die area affecting the local power supply.
- ❑ ii. Voltage threshold variation of the PMOS or the NMOS device.
- ❑ iii. Channel length variation of the PMOS or the NMOS device.
- ❑ iv. Temperature variations due to local hot spots.
- ❑ v. Interconnect metal etch or thickness variations impacting the interconnect resistance or capacitance.

上述PVT变化称为**片上偏差 (OCV, On-Chip Variations)** 这些变化能够影响芯片不同部分的**走线延迟**和**单元延迟**

由于OCV会影响时钟和数据路径，时序验证需要根据**PVT条件**对其进行建模，使得发射 (Launch) 和捕获 (Capture) 路径略有不同

静态时序分析 (STA) 可以通过**提高或降低特定路径的延迟 (Derating)** 来包含OCV效应，通过将某些路径调整到更快或更慢，并使用这些变化来验证设计的行为

单元延迟 (Cell Delay) 或者 **走线延迟 (Wire Delay)** 都可以被降额以模拟OCV的影响

- 由于位置不同，Capture寄存器的时钟延迟可能小于Launch寄存器的时钟延迟，在进行约束时，，则可以通过对Launch网络的时钟延迟进行减小，或对Capture网络的时钟延迟进行增大——**这种分析方法就体现了OCV的影响**

在这种例子中，建立时间检查条件如下（不包含OCV设置用以调整延迟）

$LaunchClockPath + MaxDataPath \leq ClockPeriod + CaptureClockPath - T_{setup_DFF}$

这意味着：

$最小时钟周期 (Minimum Clock Period) = LaunchClockPath + MaxDataPath - CaptureClockPath + T_{setup_DFF}$

- 引入OCV设置-时序减免 (**set_timing_derate命令 - 时序增减因子**)

将指定Path中的Delay按指定比例 (dreate) 放大或缩小

在进行建立时间检查时launch clock path 以及data path上的延迟已经是所有条件下最差的delay了，没有必要再加大延迟，但是WC条件下capture clock path上的delay肯定不是最小的，因此需要加快

```
# Date arrival time即data path和launch clock path需要使用 -late
选项，使得路径变慢。
# Date require time即capture clock path需要使用-early 选项，加快路
径延迟
# 在建立时间检查时，不需要增加Launch网络的延迟，只减小Capture的延迟即可
set_timing_derate -early 0.8      # 使路径变快
set_timing_derate -late 1.0      # 路径延迟不动
# -cell_delay    指定对Cell的延迟做约束
# -net_delay     指定对Net延迟做约束
# -cell_check    指定对Cell的时序检查做约束 —— 将建立时间或保持时间（检
查）约束至指定百分数
```

Derate the minimum/shortest/early paths by -20% and derate the maximum/longest/latest paths by $+10\%$.

□ Long path delays (for example, data paths and launch clock path for setup checks or capture clock paths for hold checks) are multiplied by the derate value specified using the -late option.

□ Short path delays (for example, capture clock paths for setup checks or data paths and launch clock paths for hold checks) are multiplied by the derate values specified using the -early option.

If no derating factors are specified, a value of 1.0 is assumed

■ 公共路径悲观问题

Launch时钟和Capture时钟在路径上有重合，在设置derate时对这段路径同时放大和缩小，产生了公共路径悲观 (CPP, Common Path Pessimism)，需要在分析中被移除，则引入CPPR (Common Path Pessimism Removal)

$CPP = LatestArrivalTime@CommonPoint - EarliestArrivalTime@CommonPoint$

最后将CPP作为悲观度从余量计算中减去

加入OCV，即考虑timing derate以后，时序会变得恶劣一些，从而也会降低整个design的工作频率

2. 保持时间检查 (Hold Time Check)

对于保持时间检查, 希望使Launch Clock Path和数据 Path减小, 将Capture Clock Path增加, 提高时序检查的严苛性

Data require time中的capture clock path使用-late选项, 使路径变慢。

Data arrival time中的data path和launch clock path使用-early选项, 使路径加快

◦ 引入OCV (OCV for Hold Checks)

Hold Time Check条件

$$LaunchClockPath + MinDataPath - CaptureClockPath - T_{hold_DFF} \geq 0$$

```
set_timing_derate -early 1.0    # 在Hold Check中 Launch路径Delay保持不变
set_timing_derate -late 1.2     # 在Hold Check中增加 Capture路径Delay, 以提高时序检查的严苛性
```