

CTS学习



时钟树综合问题：给定时钟单元分布，构建时钟网络（树）以实现时钟源到达各个寄存器的时序需要，期间需满足给定的各类约束，并尽可能节约设计资源。

常见时钟源：

1. PLL
2. DLL

主要概念：

1. clock latency（时钟延迟）

时钟信号从时钟源到达接收引脚的时间，可以分为：

- a. source latency：

时钟信号从时钟源到达模块边界上的端口的时间；

- b. network latency

时钟信号从端口到达接受引脚的时间；

2. clock skew（时钟偏差）

两个寄存器之间的时钟到达时间差异，可以按照以下方法分类：

- a. local：

两个寄存器之间有有效时序路径；

- b. global：

任何一对寄存器之间；

- c. positive：

时钟信号在后一级触发器到达的时间比在前一级触发器到达的时间要晚；

- d. negative：

时钟信号在后一级触发器到达的时间比在前一级触发器到达的时间要早；

e. usefu skew:

在时序路径中通过增加或者减少**缓冲器**手动**创建偏差**来修复时序违规；

3. duty cycle（占空比）

衡量信号在一个周期内处于高电平的时间占整个周期的百分比；

4. clock jitter(时钟抖动)

信号时间与理想时间的偏差；

5. Clock Gating(时钟门控)

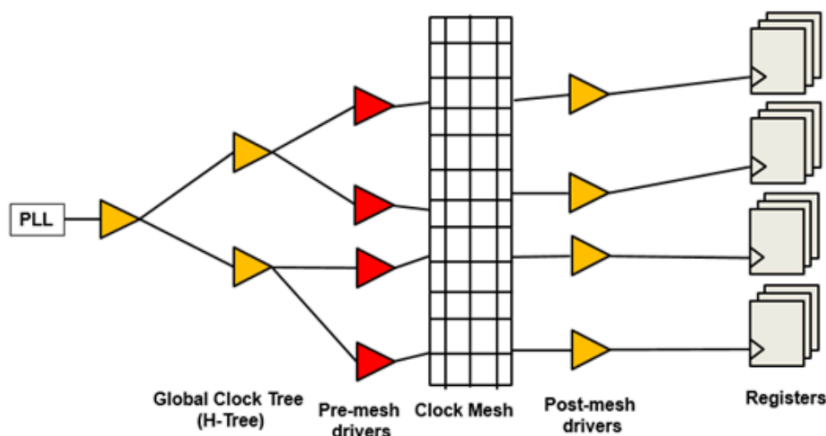
关闭空闲寄存器的时钟，可分为：粗粒度、中粒度、细粒度；

时钟树结构：

常见树形有：

- H-tree:
- X-tree
- Pi-tree
- Fish-tree
- 时钟网格

常用于高频架构，例如CPU和GPU；



优点：时钟偏移最小

缺点：功耗高、布线资源高

CTS流程:

1、输入

- 布局后网表
- LEF和Tech LEF文件
- DEF文件
- LIB
- SDC文件(标准设计约束文件): 主要内容是时序约束

2、步骤

1. Clustering
2. Balancing
3. 时钟树布线
4. 后处理

3、输出

- CTS之后的网表
- CTS DEF
- 时序报告 (建立、保持)
- 时钟偏差和延迟报告

