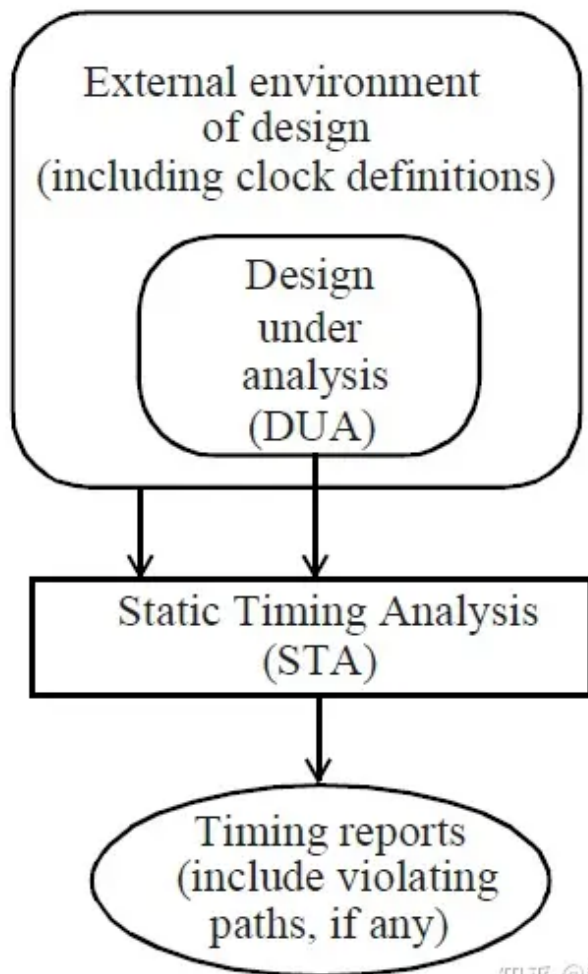


静态时序分析和优化

STA功能：



知乎 @赵俊军

- DUA:待分析的设计，通常使用硬件描述语言；
- 外部环境：包括时钟定义，通常用SDC（时序约束）文件；
- 时序报告：显示路径时延的属性；
- STA:验证设计的时序；

在时序分析前，首先要对芯片的物理版图设计进行包括电阻、电感（以及互感）、电容参数（RLC）的提取，再进行延时计算。



RC: 寄生电阻 (R) 寄生电容 (C)

芯片设计中的延时由**器件延时**和**互连线延时**两部分构成;互连线延时占**主要部分**;

静态时序分析贯穿于设计过程的各个阶段：逻辑综合，布局、时钟树综合、布线。

在不同阶段分析的目的都是为了**检查时序是否达到要求**；但是在不同阶段，准确性不一样：

在**逻辑综合**阶段，互连线参数是由**线负载模型**（wireload model）**WLS**来粗略表示的，与实际结果会相差很大；



此阶段使用理想时钟，具有零时延；

在**布局**阶段，受到I/O和逻辑模块布图的约束，互连线参数可以用比WLM稍为准确的近似值表示；

在**时钟树综合**阶段，设计的时序关键部分即与时钟相关的互连线参数已很接近最终结果了；



此阶段完成时钟树的建立，此时时钟不再是ideal clock（理想时钟）；而是propagate clock（传播时钟），具有延时；

全局布线阶段，简化的布线用于估计布线长度，无法考虑**耦合效应**带来的影响；

详细布线后，互连线**RC**最终固定成为实际值。只有在完成布线后，提取的互连线参数在经过延时计算后才是设计的实际结果；

时序约束文件：.SDC

时序分析的根本目的是为了检查在时钟的控制和要求的约束下，与其相关的数据能否符合时序要求被记录存储下来，这种时序检查就是一对常说的建立时间（setup)时序和保持时间（hold)时序。

时序路径：

- I2C(clock)
- C2C
- C2O

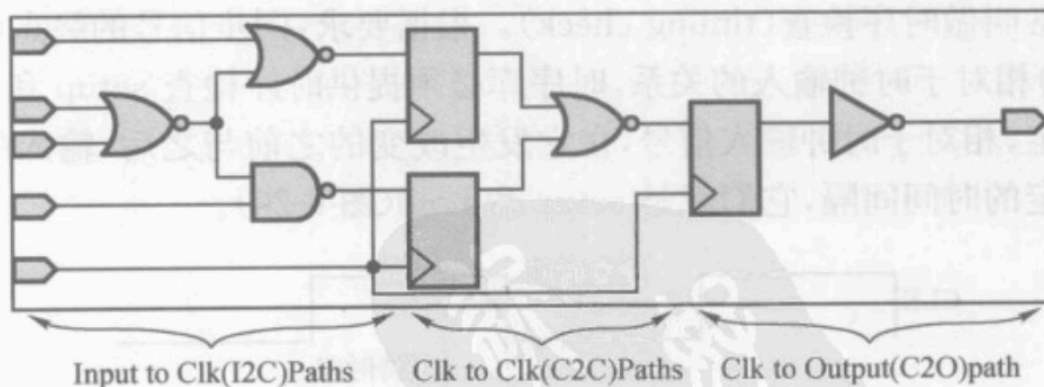


图 6-18 时序的起点、终点和时序路径

时序分析有若干种类型。建立 (setup) 和保持 (hold) 是两种普通类型的时序分析，具体分析时也常常叫做**时序检查** (timing check)

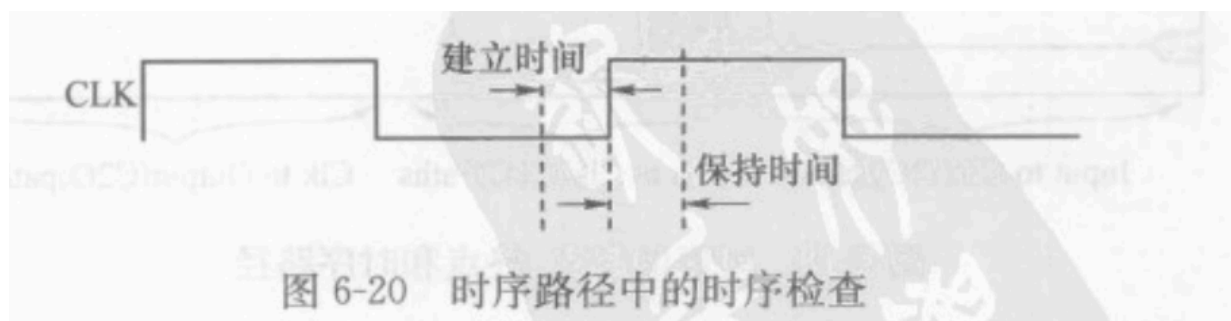
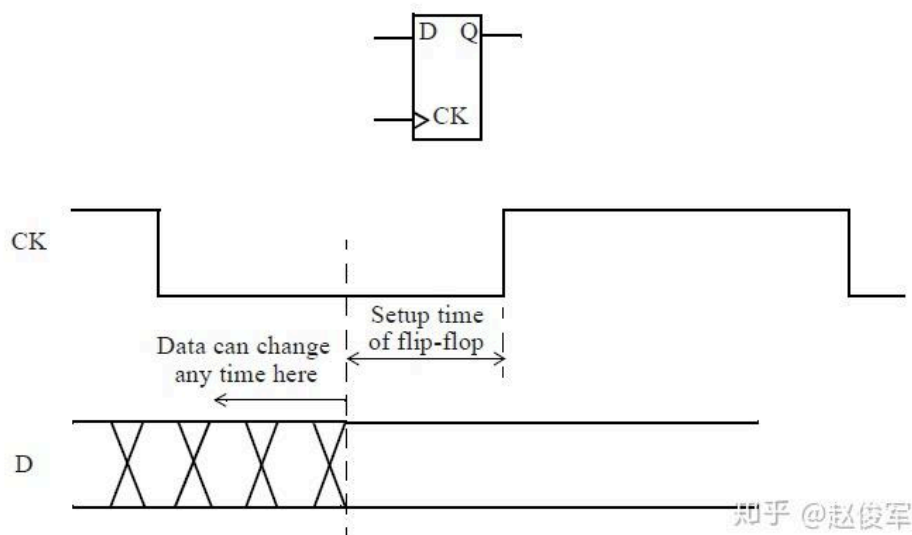


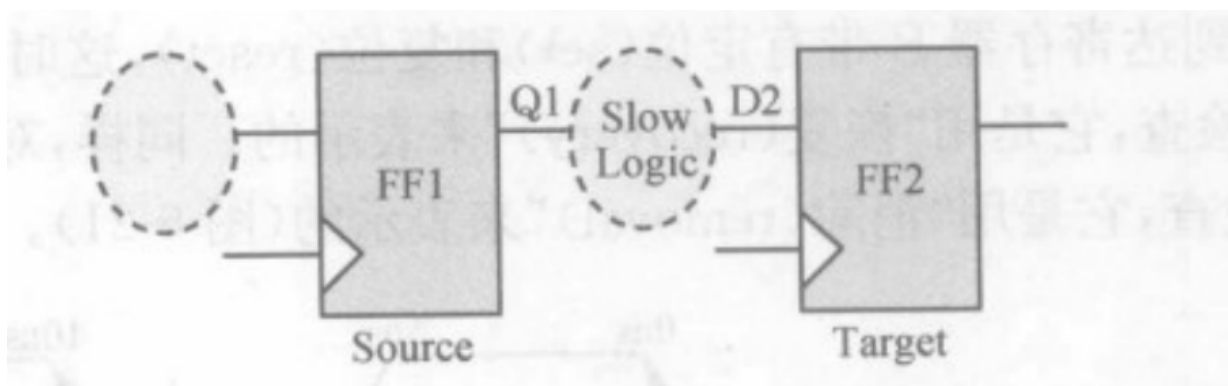
图 6-20 时序路径中的时序检查

- **setup**: 在时钟有效沿到达触发器前，数据应在一定时间内保持稳定，确保数据可靠地被捕获到触发器中；

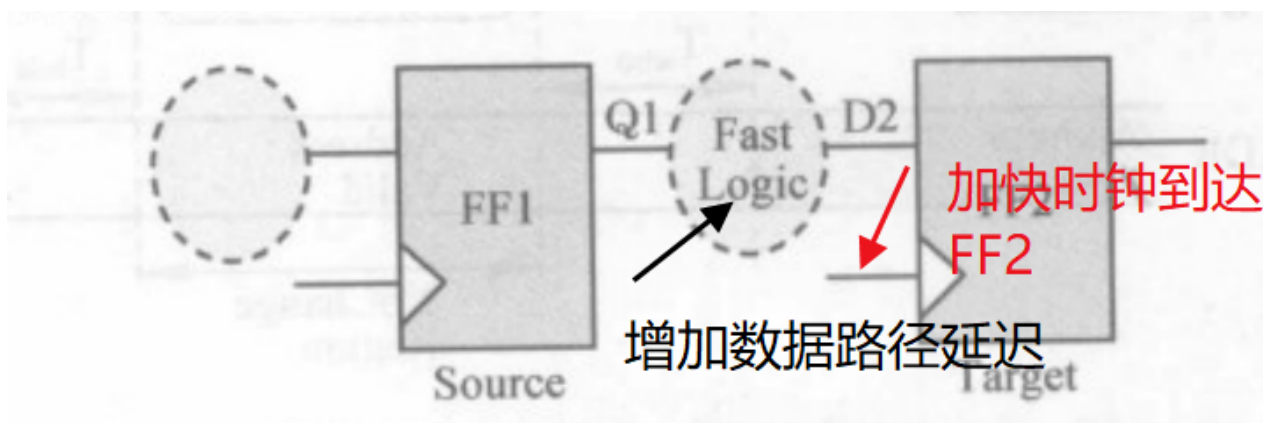


- **hold**: 在时钟的有效沿之后的指定时间段内，被锁存的数据应保持稳定；

出现setup违例的解决方法：时钟变慢(增长周期)、缩短数据路径的延迟；



出现hold违例的解决方法：增长数据路径的延迟、加快时钟到达FF2。



时序优化方法：

原地优化（IPO）

可以多次进行，例如在时钟树综合前后、布线前后进行；

IPO的特点是它仅仅对当前设计网表进行逻辑优化，它不去对网表做逻辑重组（logic structuring）

因此，经过IPO后的设计网表在后期做逻辑等效检查（LEC）时则很简单；

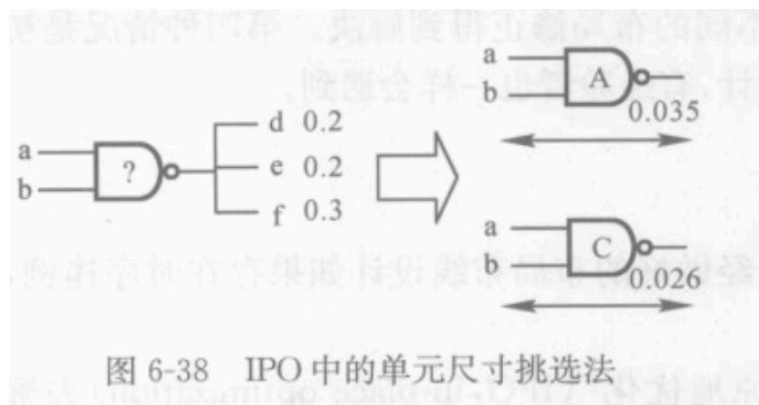
适用于：

1. 处理setup和hold违例
2. 处理设计规则（DRV）违例

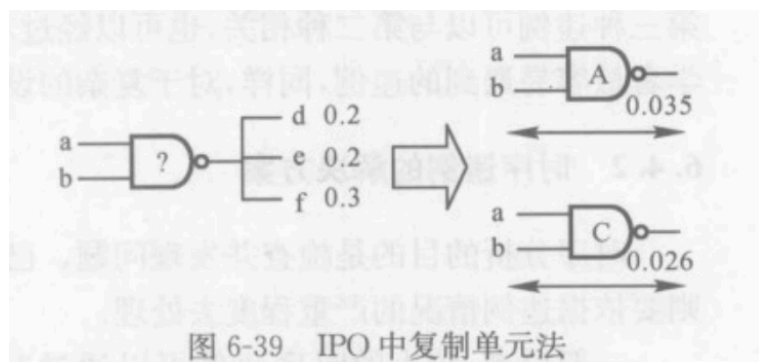
设计规则违例通常包括3种：最大负载、最大扇出和最大转换（时间）

原地优化方法：

- 1、re-sizing：挑选并替换驱动能力大小不一样的逻辑单元



2、克隆：复制一个逻辑单元去分担负载



3、添加缓冲器

