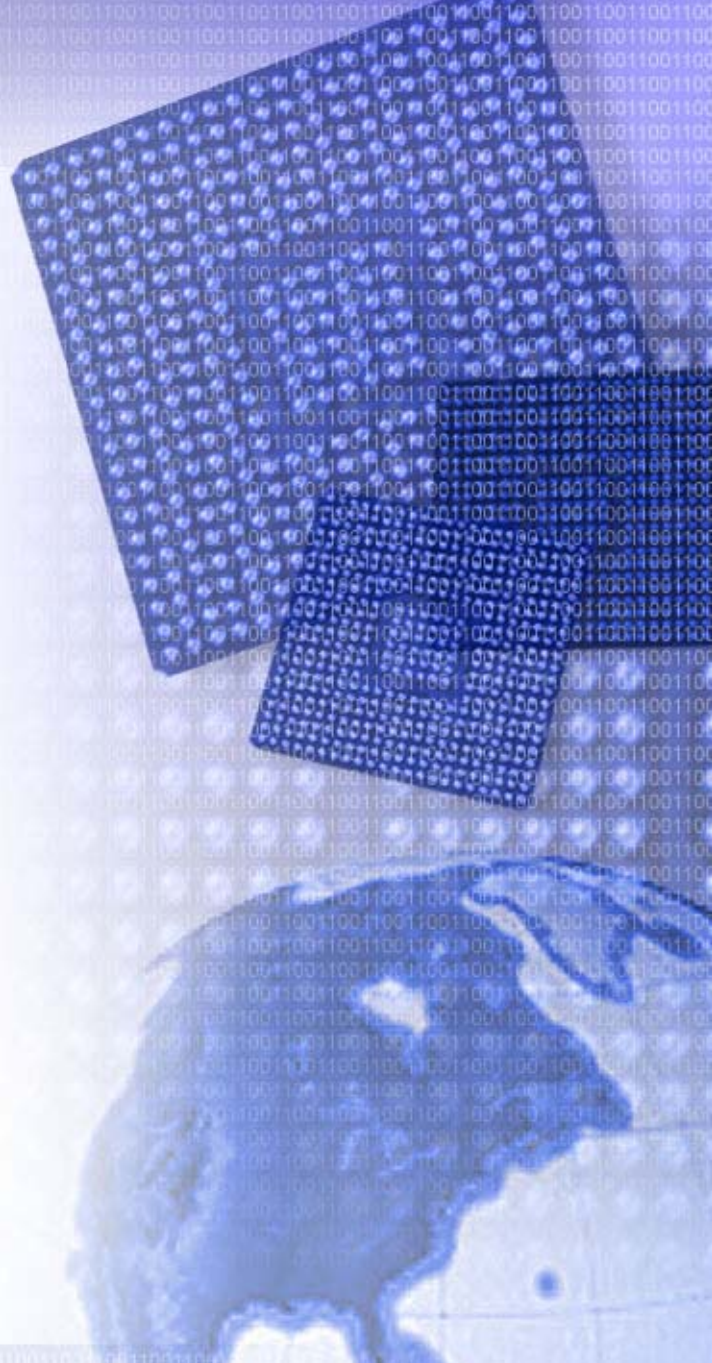


全局时序约束



目标

完成本部分内容的培训后，你将能够：

- 为简单的同步设计加全局时序约束
- 利用Constraints Editor（约束编辑器）指定全局时序约束



要点



- 简介
- 全局约束
- 约束编辑器
- 总结

时序约束和项目的关系

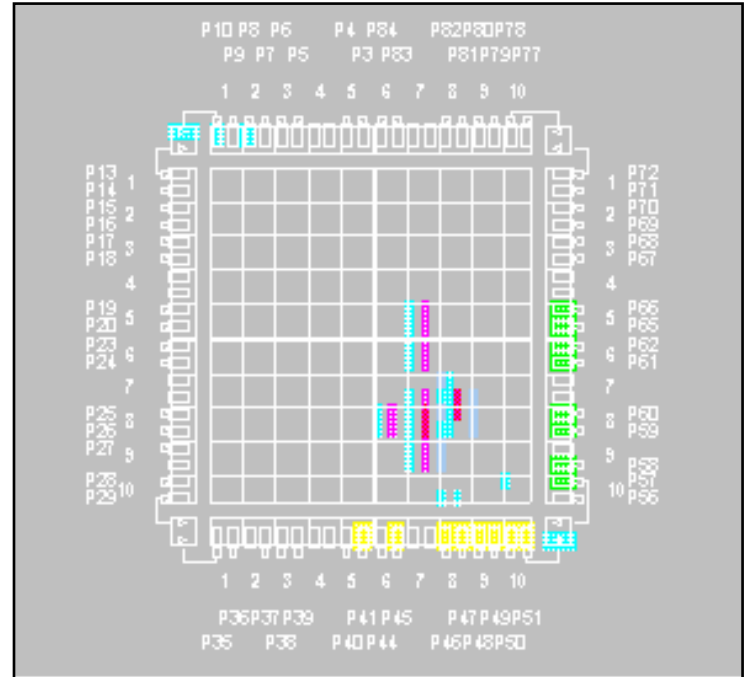
时序约束对项目有什么影响？

- 实现工具不会试图发现能够获得最佳速度的布局和布线方式
 - 相反，设计实现工具试图满足您设定的性能目标
- 性能目标通过时序约束来体现
 - 时序约束提高设计性能的途径是将逻辑放得尽可能近，从而使用尽可能短的布线资源
 - 请注意，当我们讨论约束编辑器时，我们在这儿指的是Xilinx Constraints Editor



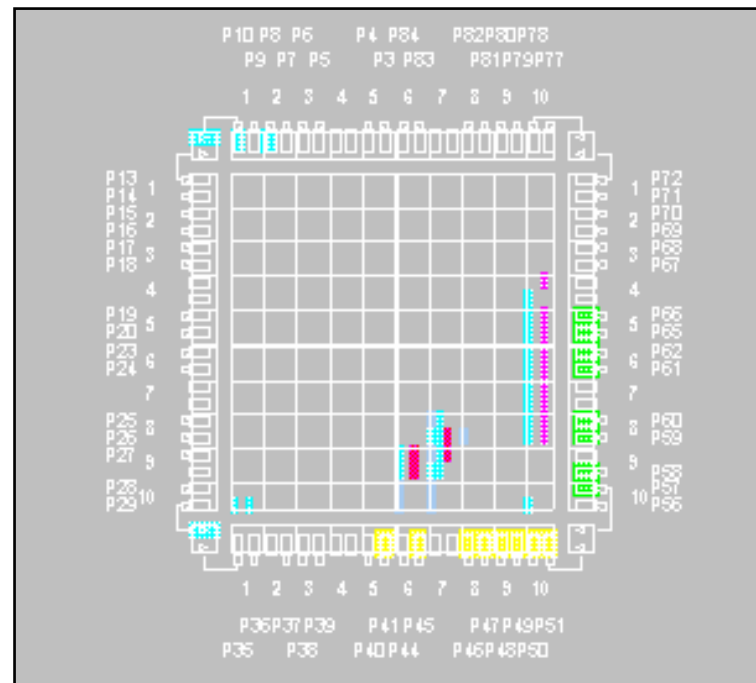
没有时序约束

- 这一设计没有时序约束和引脚分配
 - 请注意布局 and 引脚的逻辑结构
 - 这一设计的最大系统时钟频率为50 MHz



有时序约束

- 这是相同的设计，但在约束编辑器中输入了三条全局时序约束
- 这一设计的最大系统时钟频率为60 MHz
- 注意逻辑的大部分都布置在靠近器件边沿的地方，也就是靠近引脚的地方



有关时序约束的更多内容

- 应当用时序约束来定义您的性能目标
 - 过紧的时序约束将会延长编译时间
 - 不实际的约束会导致设计实现工具停止运行
 - 利用综合报告或映射后静态时序报告来判断约束是否可行
- 在设计实现后，察看布局布线后静态时序报告判断是否达到预定的性能目标
 - 如果约束没有满足，利用时序报告确定原因



路径端点

- 有两种类型的路径端点：
 - I/O 引脚
 - 同步单元（触发器、锁存器和RAM）
- 创建一条时序约束的过程包括两步
 - 步骤1：创建路径端点组
 - 步骤2：指定组之间的时序要求
- 全局约束使用缺省路径端点组
 - 所有触发器、所有I/O引脚等



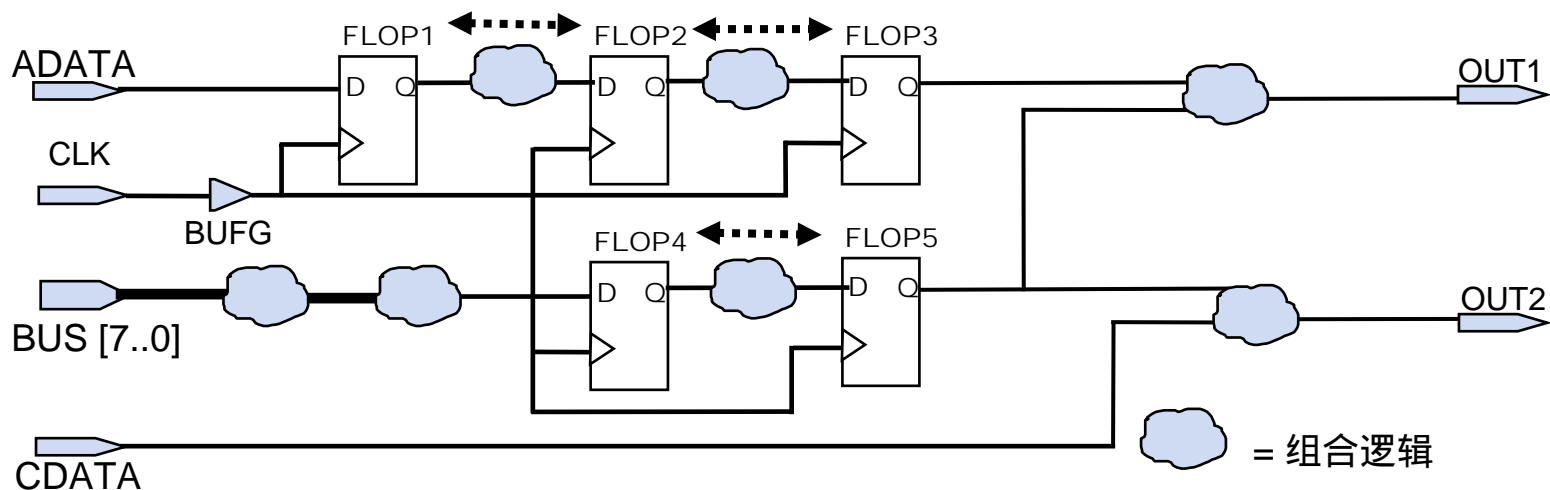


小测验



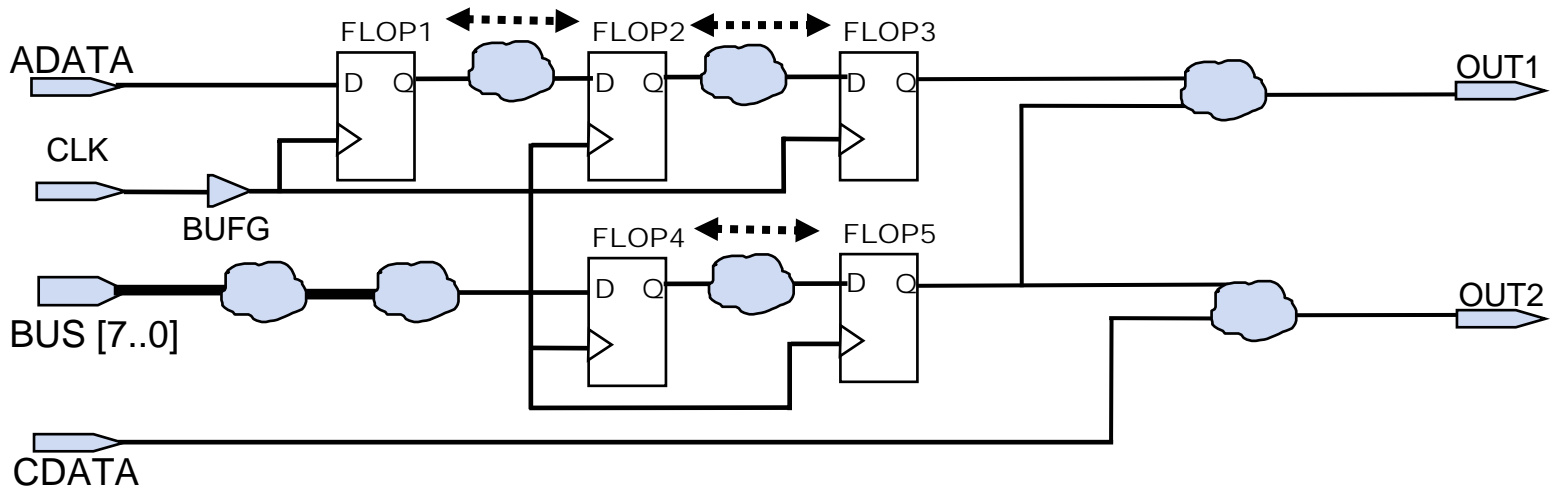
复习问题

- 一条全局约束可覆盖多条延迟路径
- 如果箭头所指的路径是约束路径，该电路中的约束路径的端点是什么？
- 本电路中的所有寄存器有什么共同之处？



答案

- 本电路中的路径端点是什么？
 - FLOP1, FLOP2, FLOP3, FLOP4, 和 FLOP5
- 本电路中的所有寄存器有什么共同之处？
 - 它们的时钟信号相同。参考该时钟网络建立的约束可以覆盖到该设计中所有寄存器之间的路径



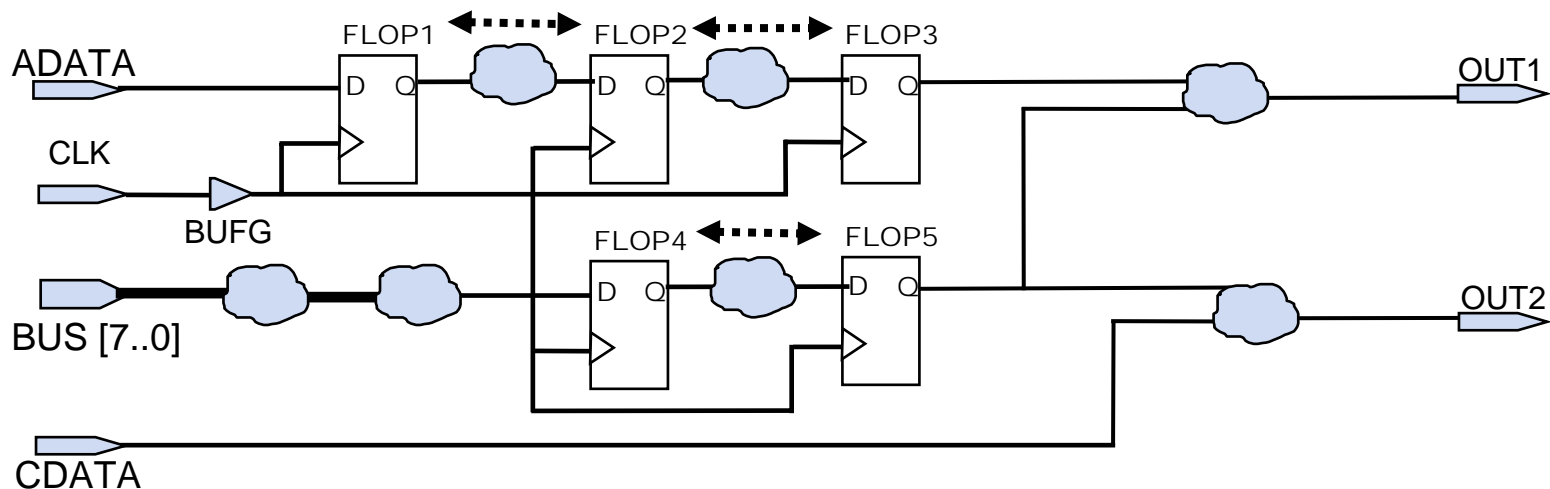
要点



- 简介
- **全局约束**
- 约束编辑器
- 总结

PERIOD约束

- PERIOD (周期) 约束适用于采用同一时钟参考网络的同步元件间的路径
- PERIOD约束不适用于从输入引脚至输出引脚 (纯组合)、从输入引脚至同步元件, 或从同步元件至输出引脚的路径

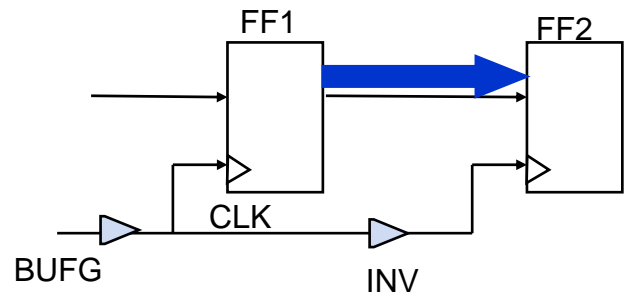


PERIOD约束的特点

- PERIOD约束应该使用最准确的时序信息，工具会自动考虑到：
 - 源和目标触发器间的时钟相移
 - 同步在时钟负沿的同步元件
 - 不对称的时钟占空比
 - 时钟输入抖动

假设：

- 50-%占空比的CLK信号
- 10ns的PERIOD约束
- 由于FF2时钟采用CLK的下降沿，因此两个触发器间路径的实际约束为10ns的50% = 5ns



时钟输入抖动

- 时钟输入抖动是时钟不确定性的原因之一
- 时钟不确定性被从下面路径中减去
 - PERIOD约束的建立路径
 - OFFSET IN约束的建立路径
- 时钟不确定性被加到下列路径中：
 - PERIOD约束的保持路径
 - OFFSET IN约束的保持路径
 - OFFSET OUT约束路径



引脚至引脚 (Pad-to-Pad) 约束

- 适用于不包含任何同步元件的纯组合延迟路径
- 始于或结束于I/O引脚的纯组合延迟路径 (经常不做约束)



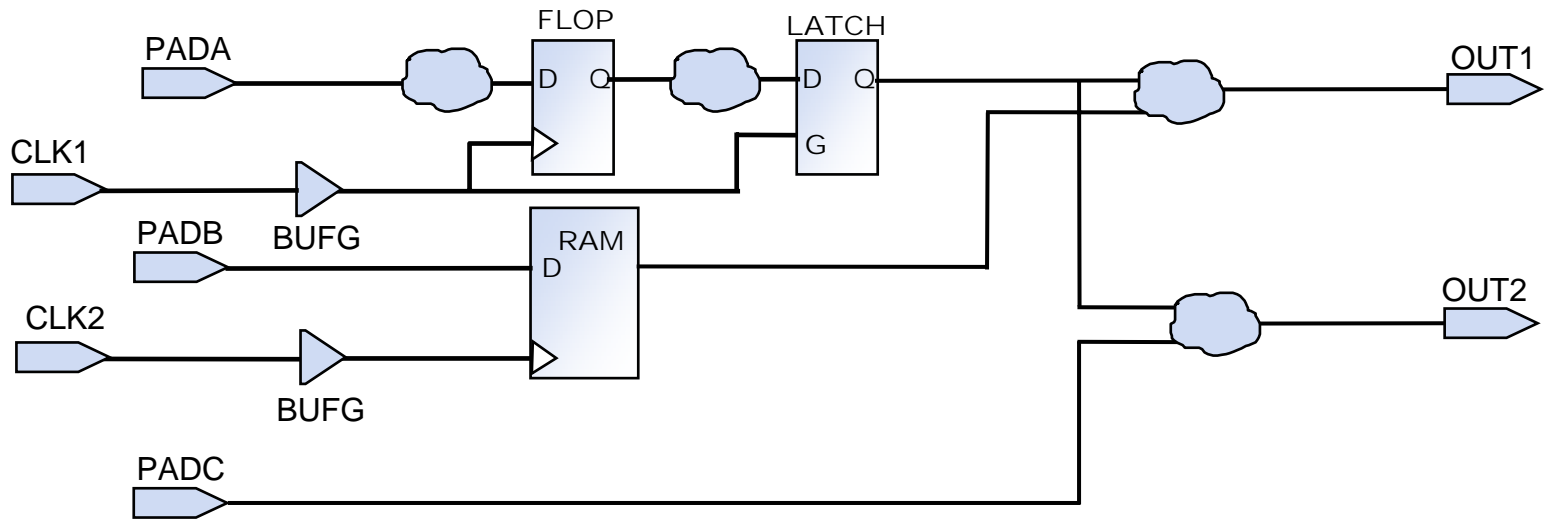


小测验



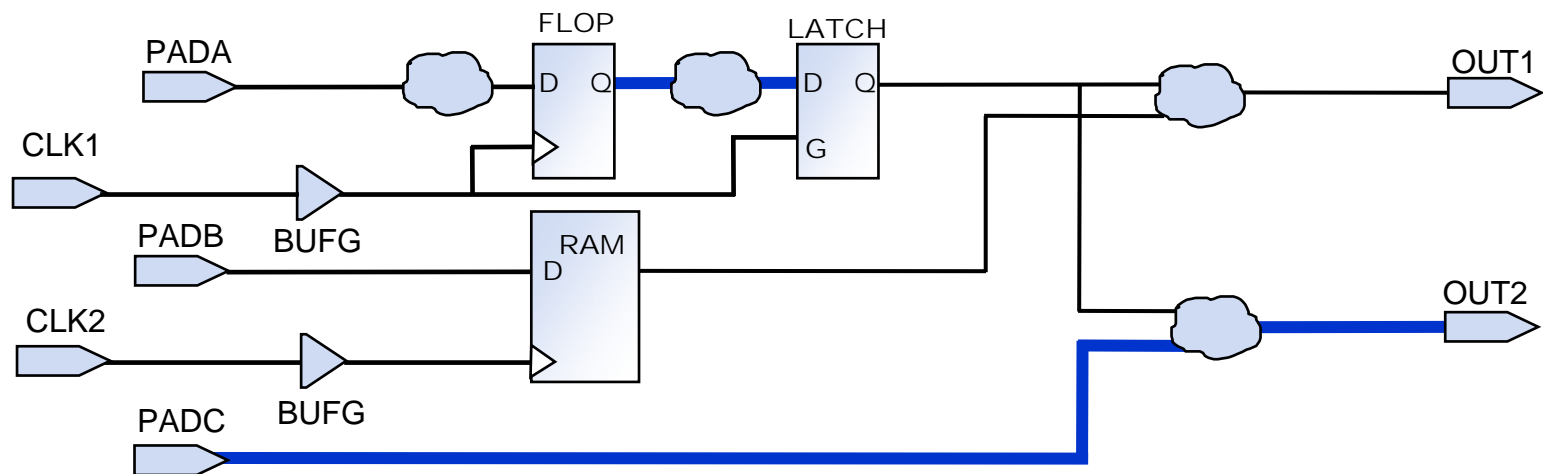
复习问题

- 哪些路径受CLK1上PERIOD约束的约束？
- 哪些路径受引脚至引脚约束的约束？



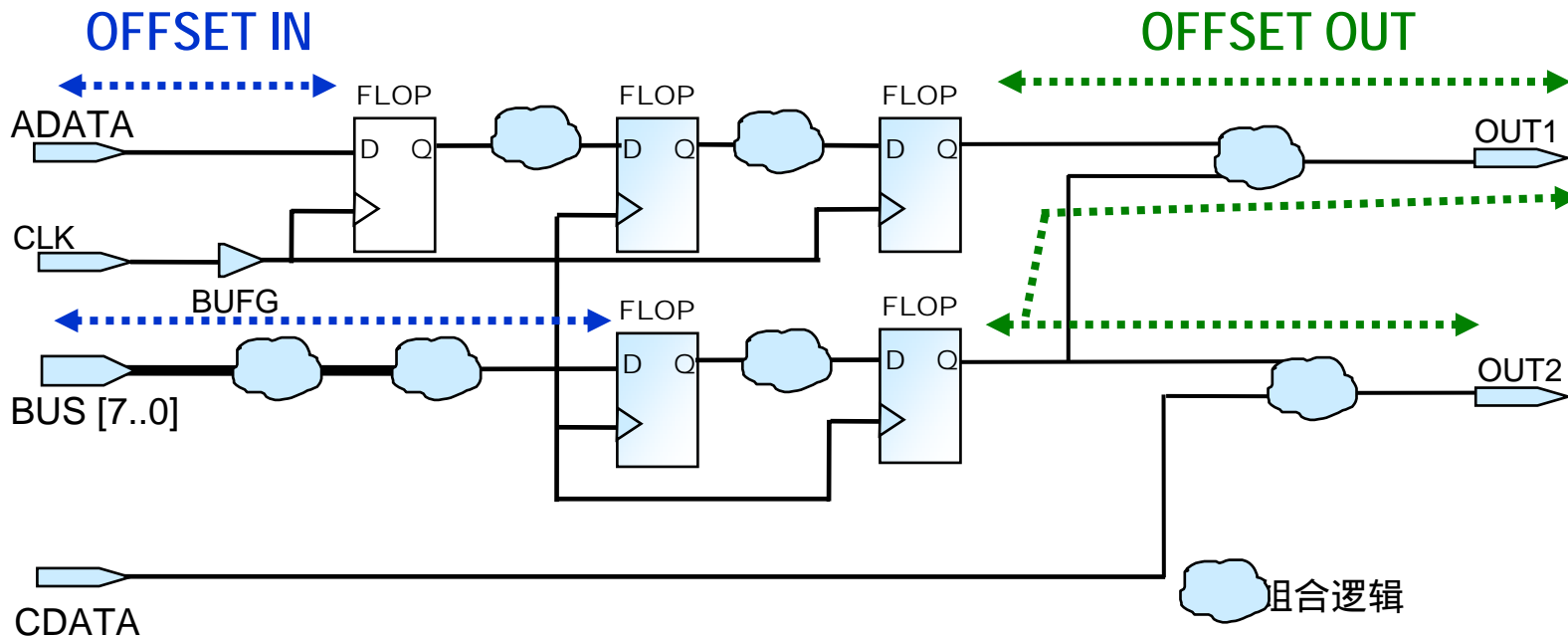
答案

- 哪些路径受CLK1上PERIOD约束的约束？
 - FLOP至 LATCH
- 哪些路径受引脚至引脚约束的约束？
 - PADC 至OUT2



OFFSET约束

- OFFSET约束适用的路径：
 - 从输入引脚至同步元件 (OFFSET IN)
 - 从同步元件至输出引脚 (OFFSET OUT)



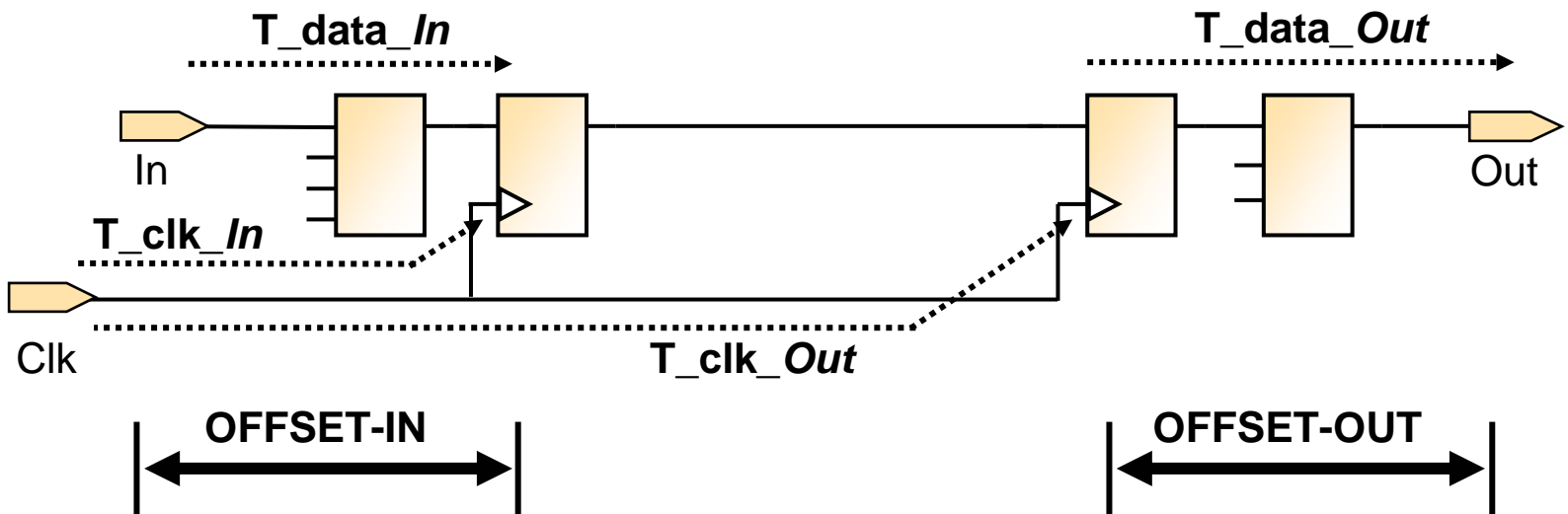
OFFSET约束的特点

- OFFSET约束会自动考虑到时钟分布延时
 - 提供最准确的时序信息
 - 增加输入信号到达同步元件的时间（时钟和数据通道是并行的）
 - 减少输出信号到达输出引脚的时间（时钟和数据通道是串行的）
- OFFSET约束还考虑到时钟输入抖动
 - 利用相关PERIOD约束所定义的抖动



时钟延迟

- 计算OFFSET约束时同时使用数据通道延迟和时钟传输延迟
 - $OFFSET\ IN = T_data_In - T_clk_In$
 - $OFFSET\ OUT = T_data_Out + T_clk_Out$



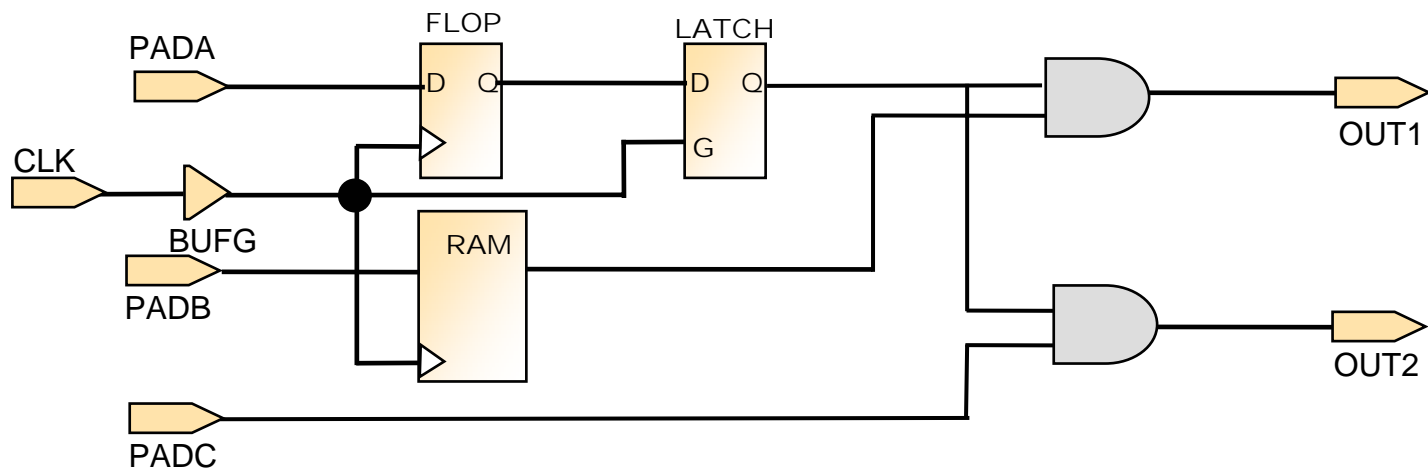


小测验



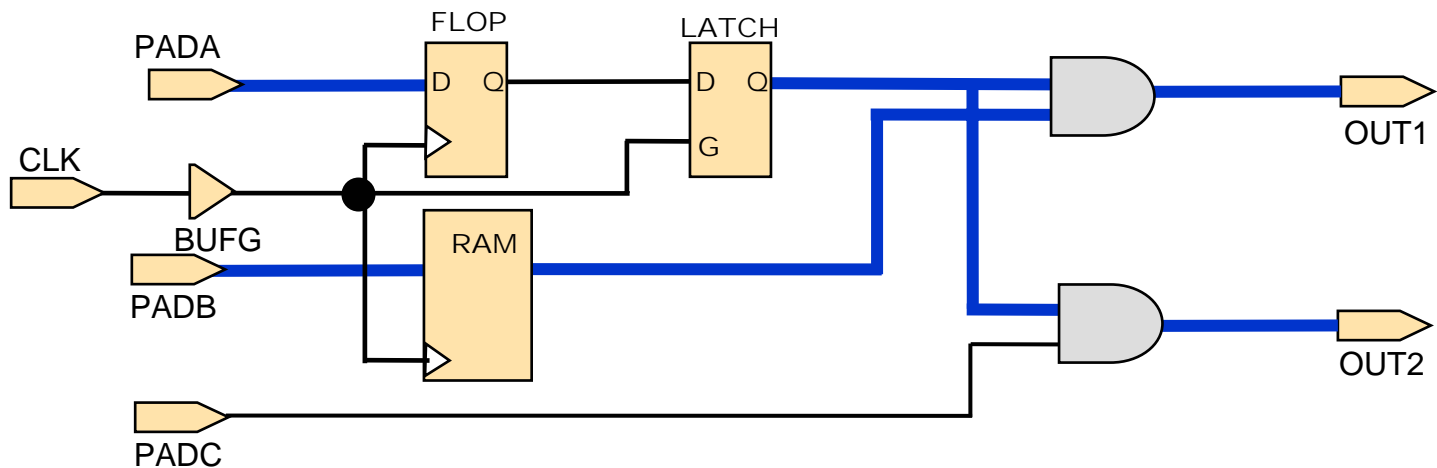
复习问题

- 本电路中哪些路径受OFFSET IN和OFFSET OUT约束的约束？



答案

- 本电路中哪些路径受OFFSET IN和OFFSET OUT约束的约束？
 - OFFSET IN: PADA 至 FLOP 和 PADB 至 RAM
 - OFFSET OUT: LATCH 至OUT1, LATCH 至OUT2, 和RAM 至OUT1



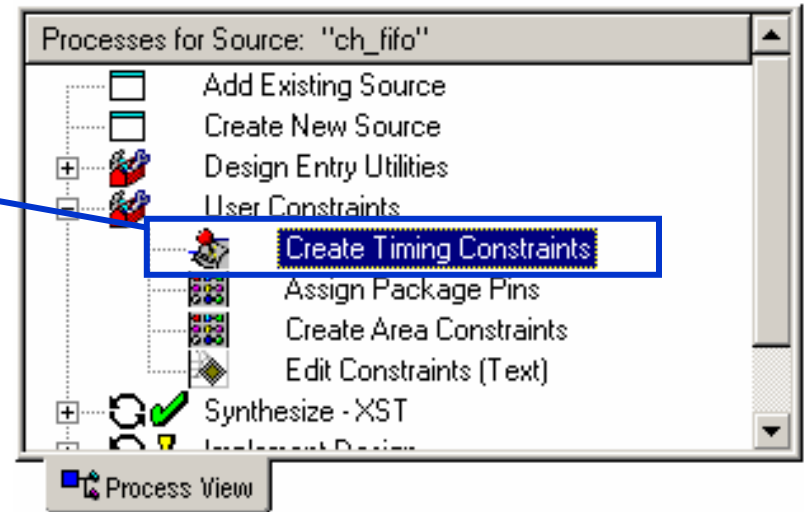
要点

- 简介
- 全局约束
- **约束编辑器**
- 总结



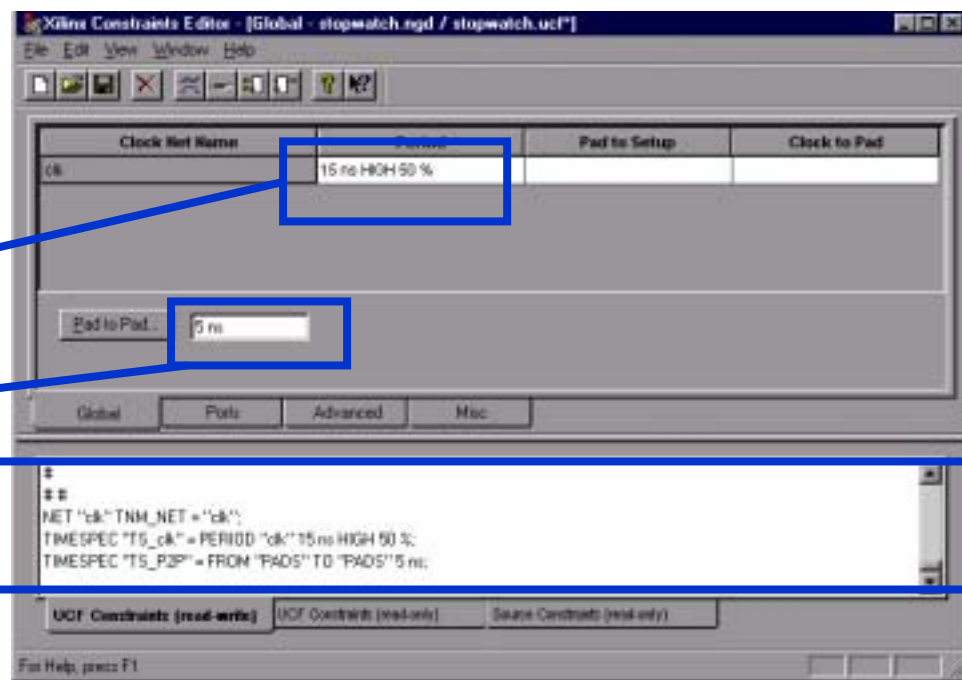
启动Constraints Editor (约束编辑器)

- 在Processes for Source窗口中
，展开User Constraints
- 双击Create Timing Constraints (创建时序约束)



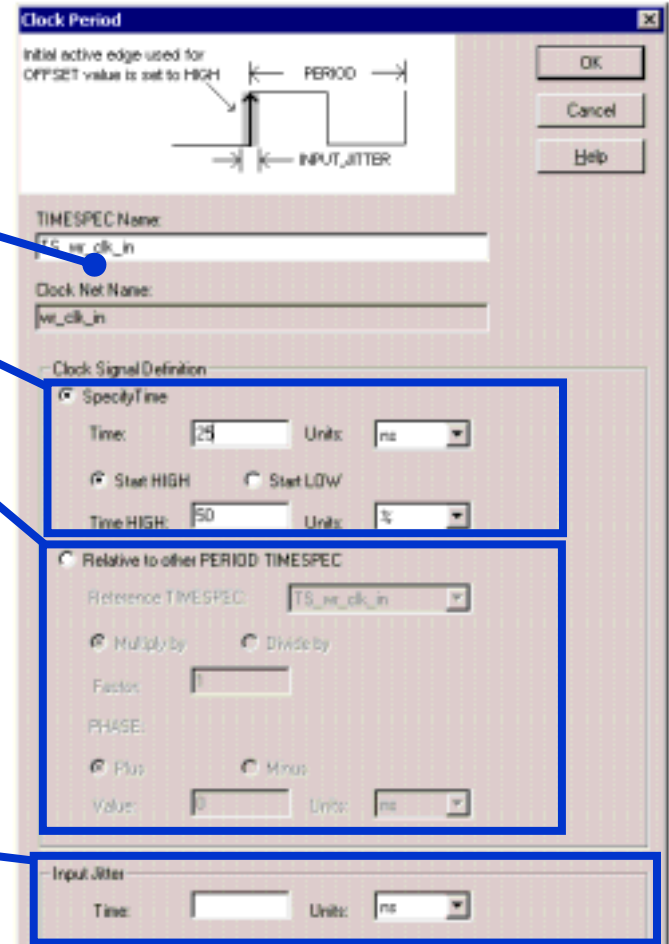
输入PERIOD和 Pad-To-Pad约束

- PERIOD和pad-to-pad约束可在Global tab下输入
- 双击这儿创建一个PERIOD约束
- 全局Pad-to-Pad约束
- 通过在文本窗口中选择约束并按下<Delete>，可以删除已建立的约束



PERIOD约束选项

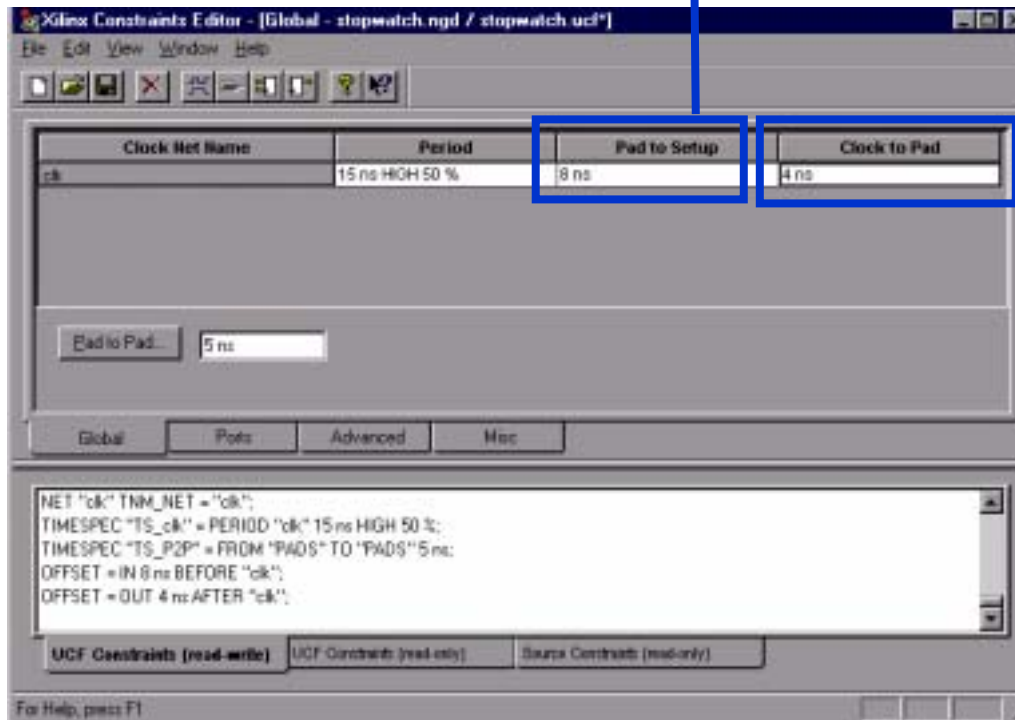
- TIMESPEC 名称
- 详细的约束数值
 - 有效时钟沿
 - 占空比
- 派生PERIOD约束
 - 对于有多个时钟信号的设计很有用
 - 可同时定义频率和相位关系
- 输入抖动



输入OFFSET约束

- 全局OFFSET IN/OUT约束也可在Global标签下建立

Pad to setup = OFFSET IN



Clock to pad =
OFFSET OUT

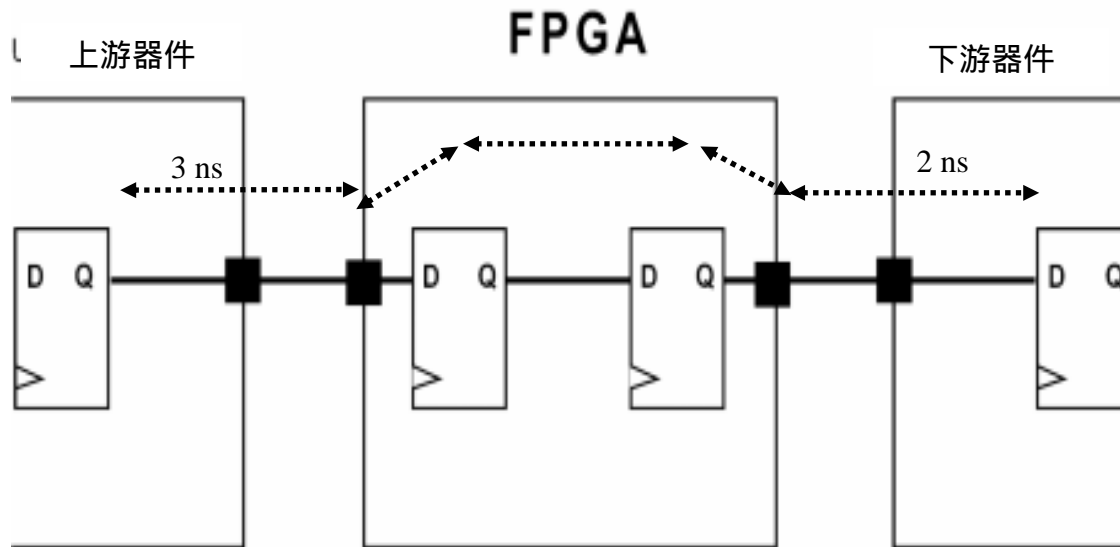
要点

- 简介
- 全局约束
- 约束编辑器
- 总结



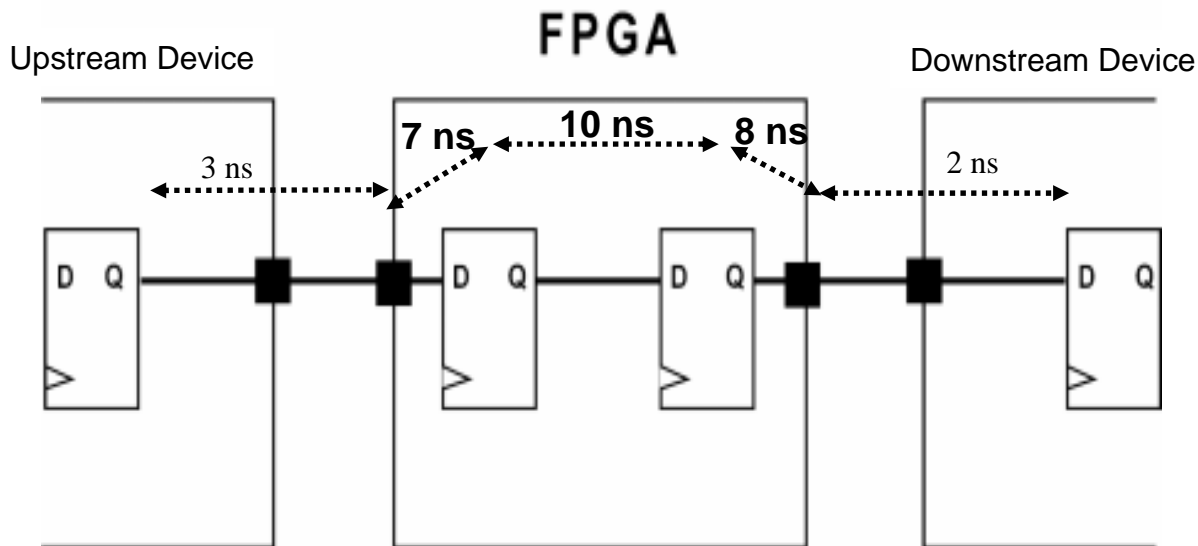
复习问题

- 对于下面的系统框图，要保证系统运行在100 MHz，需要在Constraints Editor中输入什么数值？
 - 假设器件间没有时钟相移和歪斜



答案

- 对于下面的系统框图，要保证系统运行在100 MHz，需要在Constraints Editor中输入什么数值？



- 答案 PERIOD = 10 ns , OFFSET IN = 7 ns 和 OFFSET OUT = 8 ns

总结

- 性能目标通过时序约束来体现
- PERIOD约束适用于同步单元间的延迟路径
- OFFSET约束适用于从输入引脚至同步单元以及从同步单元到输出引脚的延迟路径
- 利用Constraints Editor可创建时序约束



哪儿可以学习更多内容？

- <http://support.xilinx.com>网站上的时序问题演示文档
 - Documentation → Technical Tips → Timing & Constraints → Getting Started → The Timing Presentation
- <http://support.xilinx.com>网站上的时序改善向导（Timing Improvement Wizard）
 - 点击Problem Solver菜单

