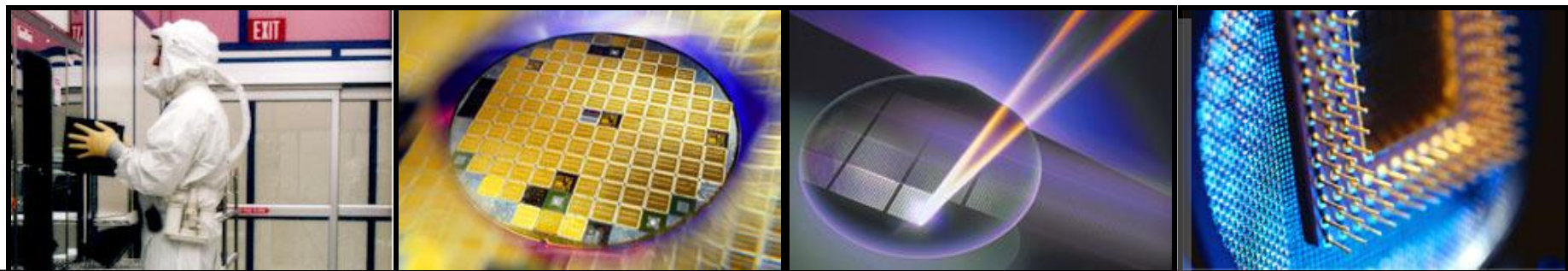




FPGA静态时序分析



课程安排

- 静态时序分析基本概念
- 寄存器的基本参数
- 时钟参数满足的条件
- 实例：电路的时序分析方法

课程安排

- 静态时序分析基本概念
- 寄存器的基本参数
- 时钟参数满足的条件
- 实例：电路的时序分析方法

时序分析概念

- 时序分析的目的

对设计进行时序的检查与分析

- 为何要进行时序分析？

器件本身固有延时

互连线的延时

时序分析的目的

- 电路的正常工作依赖于电路工作的快慢，即电路的时序特征
- 时序分析可以在芯片设计阶段，为设计者提供电路的时序特征。
- 时序分析可以为电路的正常工作提供时序保证。它可以完成对电压、温度、工艺等参数的分析。

怎么办？

- 功能是否可以和时序分开？
- 关心信号变化发生的时刻，而不是信号如何发生变化。

时序分析的类型

- 静态时序分析 - Static Timing Analysis

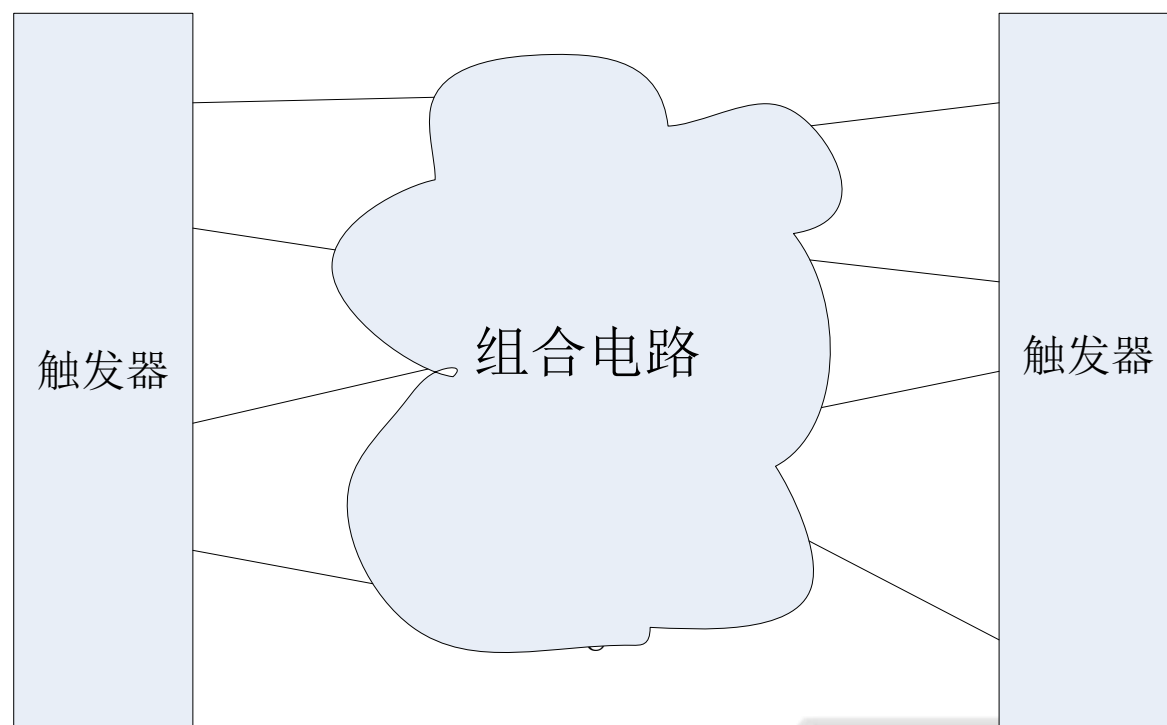
套用特定的时序模型，针对特定电路分析其是否违反设计者给定的时序限制。STA是一种穷尽分析方法，用以衡量电路性能。它提取整个电路的所有时序路径，通过计算信号沿在路径上的延迟传播找出违背时序约束的错误，主要是检查建立时间和保持时间是否满足要求，而它们又分别通过对最大路径延迟和最小路径延迟的分析得到。静态时序分析的方法不依赖于激励，且可以穷尽所有路径，运行速度很快，占用内存很少

- 动态时序分析 - Dynamic Timing Analysis

主要指的是门级仿真，这种方法主要应用在异步逻辑、多周期路径、错误路径的验证，根据输入信号的向量进行动态的时序验证，

静态时序分析的模型

- 电路是同步运行的。
- 电路被触发器分割。



基本原理

- 不考虑长时间的仿真序列
- 将信号的各种变化“折叠”到一个时钟周期
- 寻找信号最晚进入稳定的时刻
- 寻找信号最早脱离稳定的时刻
- 通过电路在极端情况下的分析保证电路在各种情况下都可以正常工作
- “静态”相对于“动态”是指没有仿真。

静态时序分析

- 定义：在不模拟的条件下，计算电路是否符合时延约束。
- 步骤：
 - 设计分解为一系列的时延路径
 - 每一条路径都计算延时
 - 检查所有的路径

静态时序分析的对象

分析的路径

- 时钟分析
- 寄存器的建立/保持时序分析
- 输入输出延时
- 节点与节点间的延时
- 不同厂家的器件的分析路径会略有不同

进行时序约束

- **非常重要 !!**
- 对设计有决定性的影响
 - 用时序约束来为设计详细指定所有的时序需求
 - 布线器按时序约束来布局布线，会在最差的路径上作最大的努力
 - 若时序不满足，以红色字体报告出来
- 类型
 - 内部节点的约束 & 外部 I/O 的约束
- 进行全局约束或者单个约束
 - 推荐全局约束

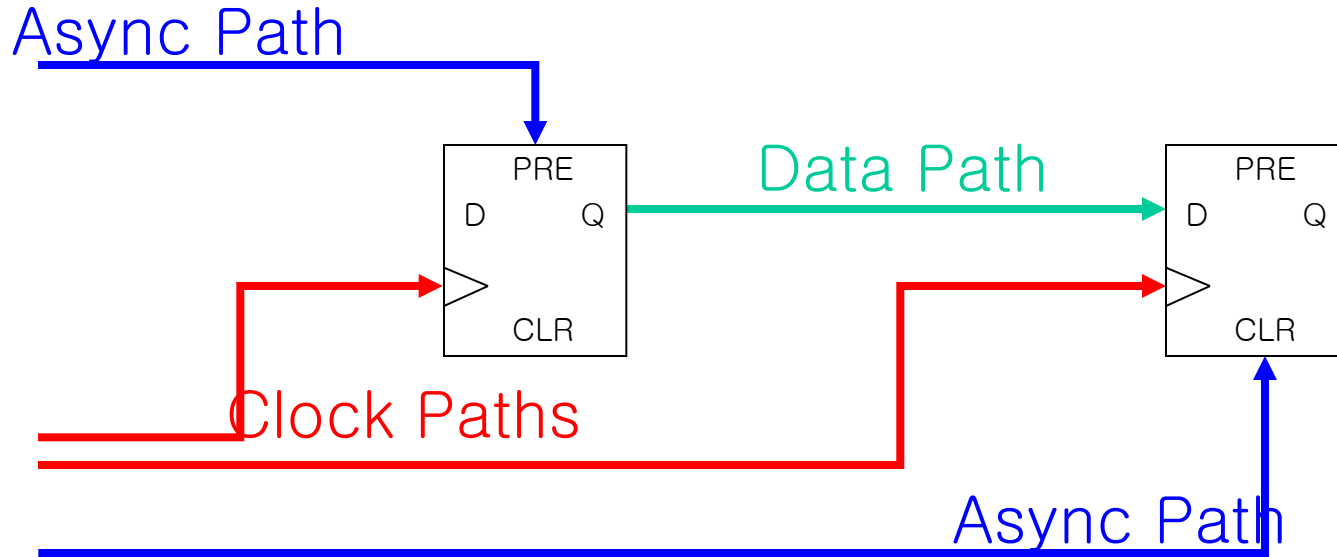
时序模型 Timing Models

- 不同的器件有不同的时序模型
- FPGA厂家提供时序模型进行路径延时的估计

课程安排

- 静态时序分析基本概念
- 寄存器的基本参数
- 时钟参数满足的条件
- 实例：电路的时序分析方法

路径和分析类型



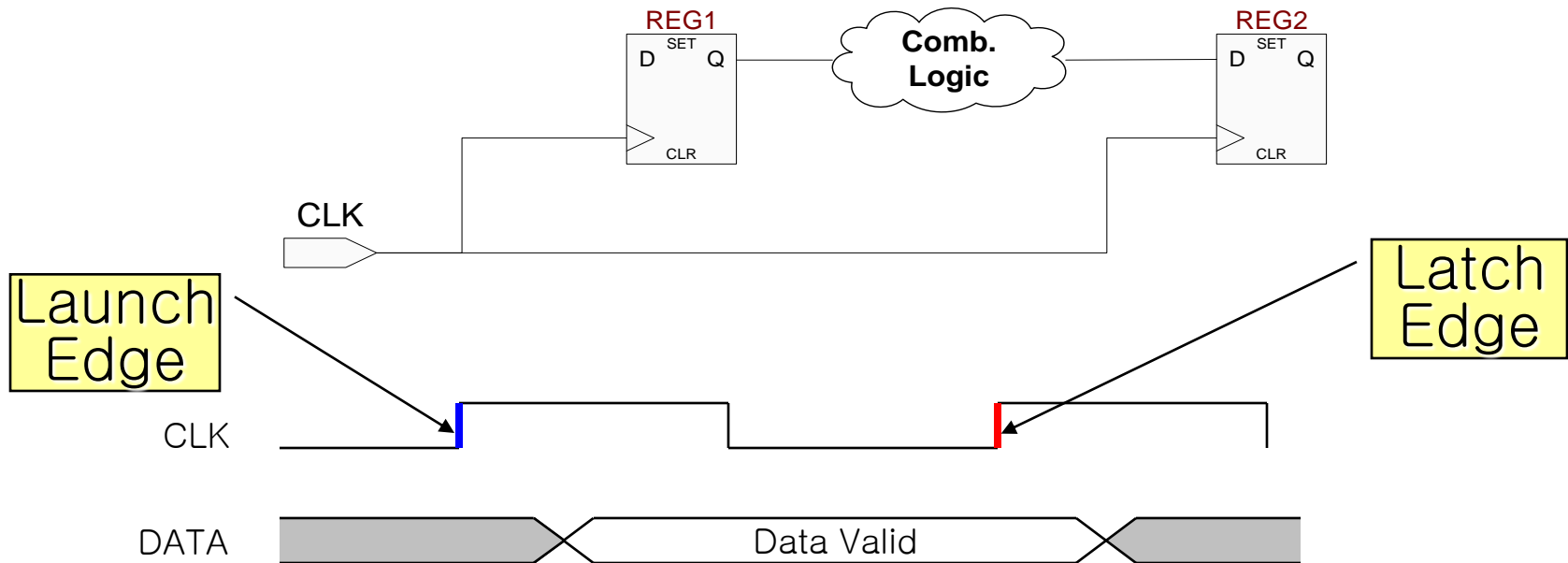
Three types of Paths:

1. **Clock Paths**
2. **Data Path**
3. **Asynchronous Paths***

Two types of Analysis:

1. Synchronous – clock & data paths
2. Asynchronous* – clock & async paths

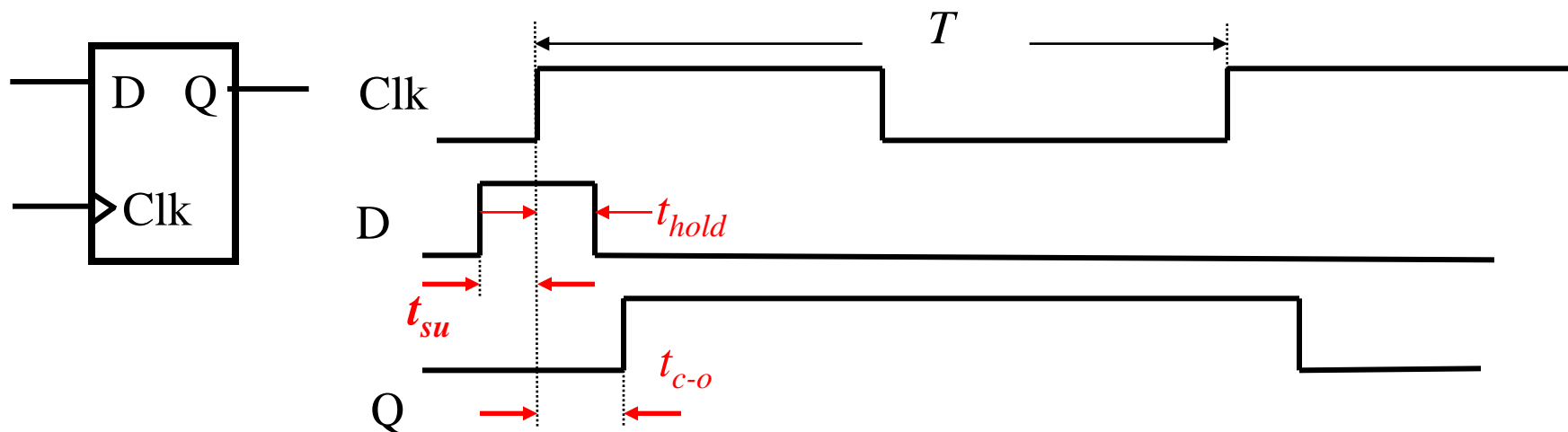
Launch & Latch Edges



Launch Edge: the edge which “launches” the data from source register

Latch Edge: the edge which “latches” the data at destination register (with respect to the launch edge, selected by timing analyzer; typically 1 cycle)

Register的参数



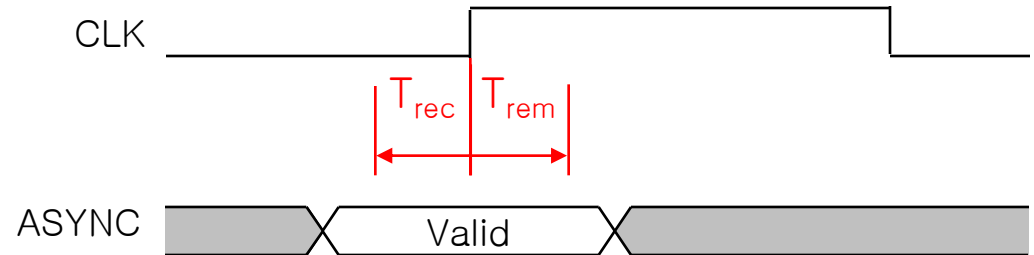
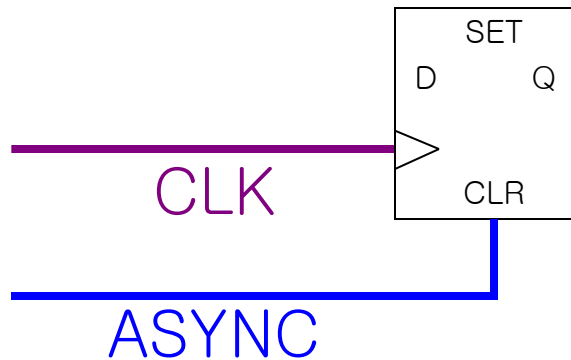
tsu:建立时间，在时钟有效沿到来之前寄存器数据输入应保持稳定的时间，它间接约束了组合逻辑的最大延时

thold:保持时间，在寄存器数据输入的引脚的数据在系统有效时钟沿到来后需要保持稳定的时间，它间接约束了组合逻辑的最小延时

tc-o:寄存器从有效时钟沿到来输出有效的最大时间

违反建立或保持时间，都会造成触发器工作异常，产生Metastability。为了可靠工作，在建立时间开始到保持时间为止的这段时间内，触发器的输入端信号不应发生变化。

Recovery & Removal

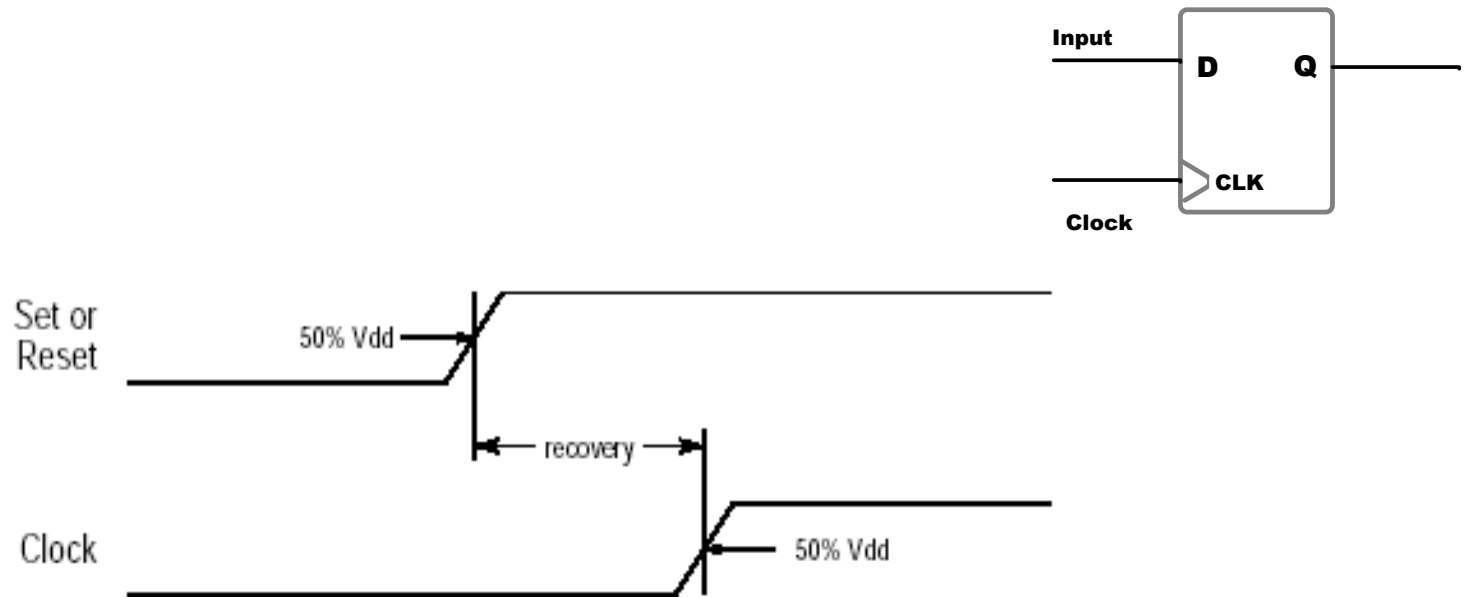


Recovery: The minimum time an asynchronous signal must be stable BEFORE clock edge

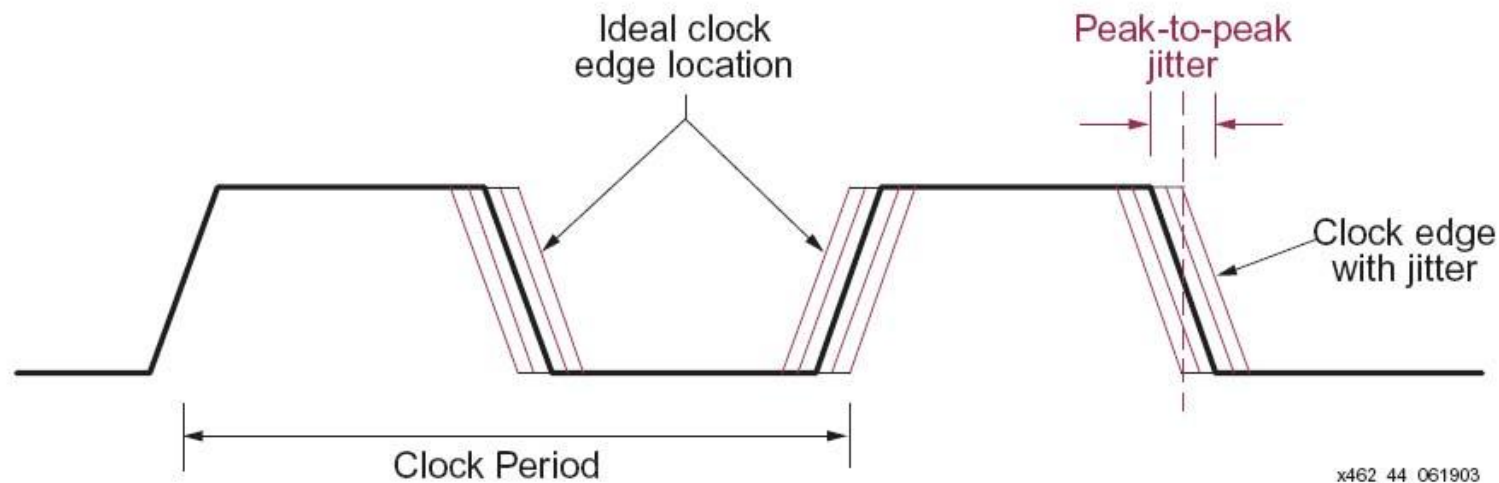
Removal: The minimum time an asynchronous signal must be stable AFTER clock edge

Recovery Time

- Recovery time – is the minimum length of time that the active low set or reset signal must remain high before the active edge of the clock to ensure correct functioning of the cell.



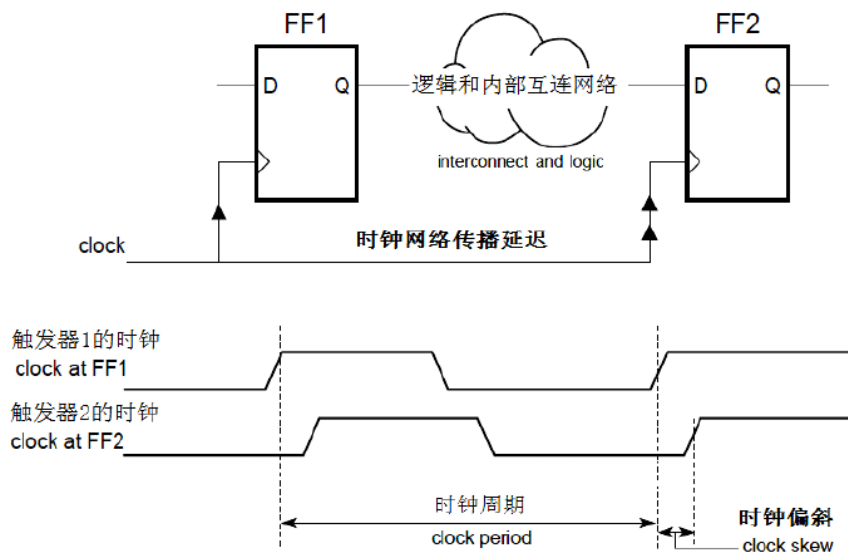
时钟不确定性



- 输入时钟抖动是时钟不确定性的原因之一(Clock Uncertainty)

时钟偏斜 (clock skew)

- 所有时序设计和分析都以时钟网络为参考。路径延迟分析必须考虑时钟质量。
- 时钟偏斜是一个时钟信号沿着同一个时钟网络到达源触发器和目标触发器的时间差。
- 所有时钟网络布线应该使用FPGA提供的专用时钟资源，否则时钟偏斜会非常严重。
- 时钟偏斜分为负时钟偏斜和正时钟偏斜，主要对时钟周期产生影响。时序分析器能够报告时钟偏斜问题。



时钟偏斜 (clock skew)

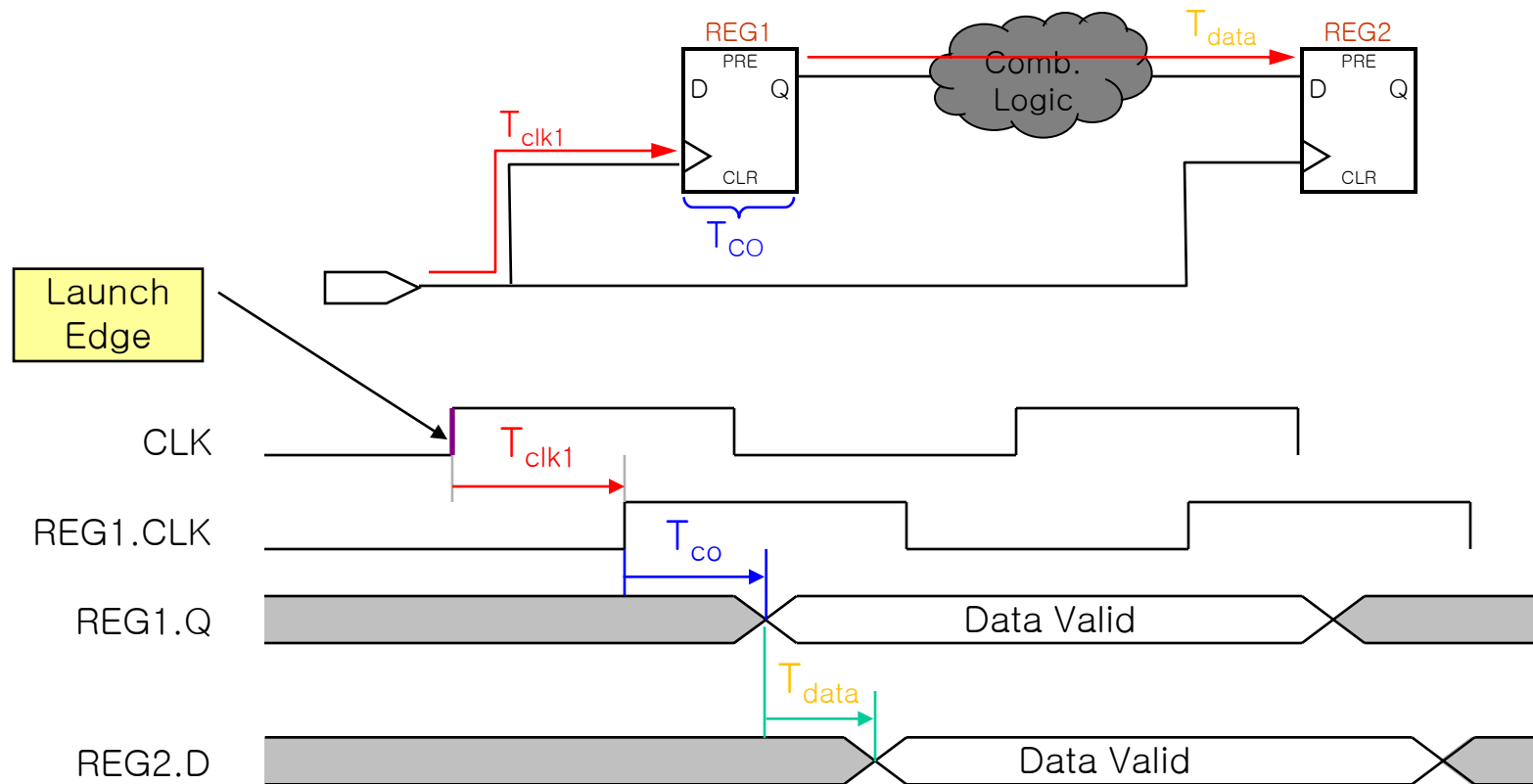
- Clock skew: Variation of in arrival time of a clock transition to flops
- Clock skew is caused by mismatches in the clock paths:
 - Unequal wire length
 - Unequal buffer delay
 - Unequal load
 - IR drop
 - In-die process variation

课程安排

- 静态时序分析基本概念
- 寄存器的基本参数
- 时钟参数满足的条件
- 实例：电路的时序分析方法

数据到达时间

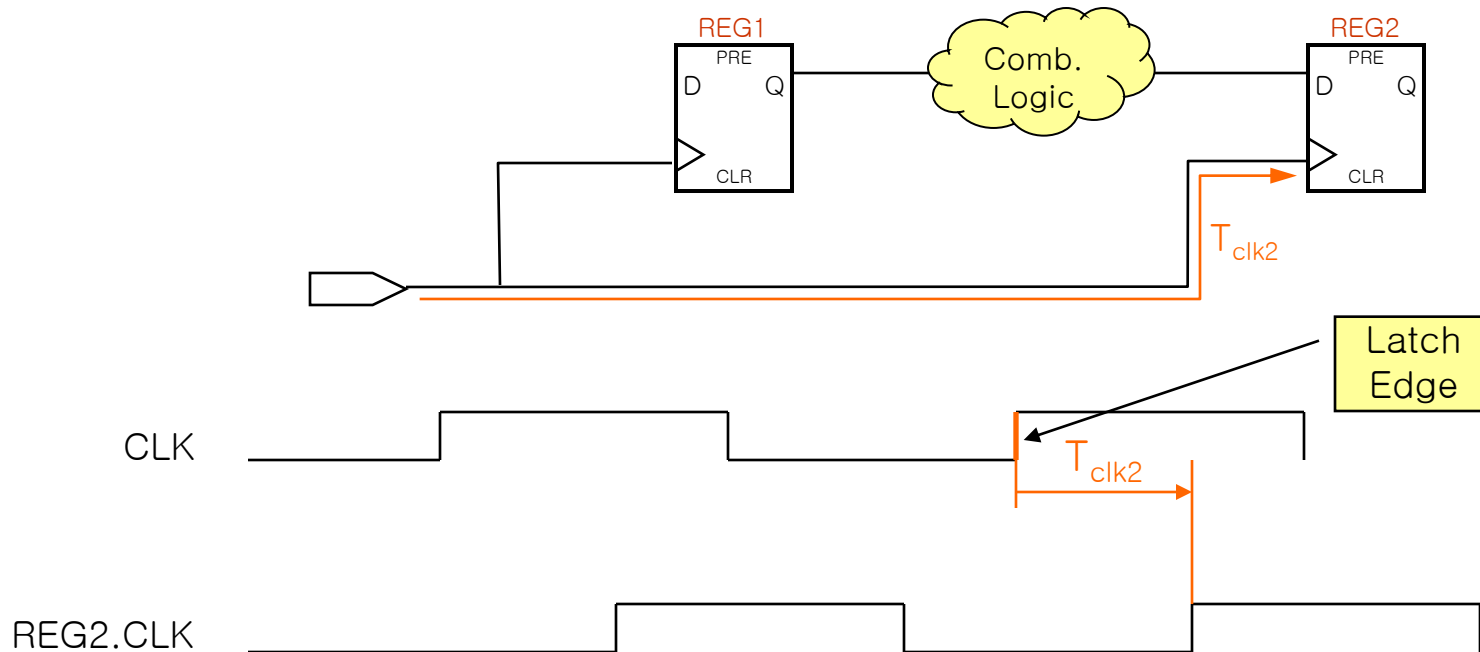
- The time for data to arrive at destination register's D input



24 $\text{Data Arrival Time} = \text{launch edge} + T_{clk1} + T_{co} + T_{data}$

时钟到达时间

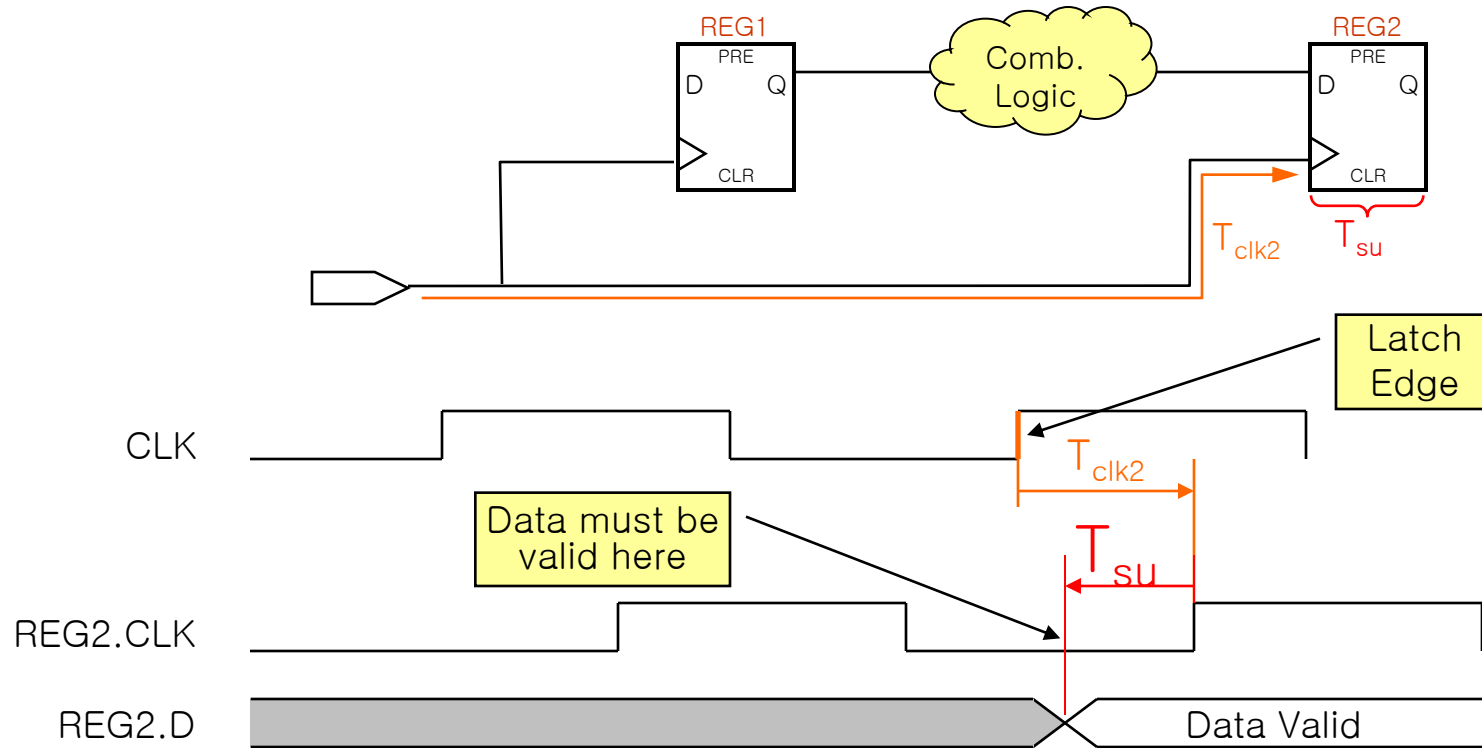
- The time for clock to arrive at destination register's clock input



$$\text{Clock Arrival Time} = \text{latch edge} + T_{clk2}$$

数据要求时间- Setup

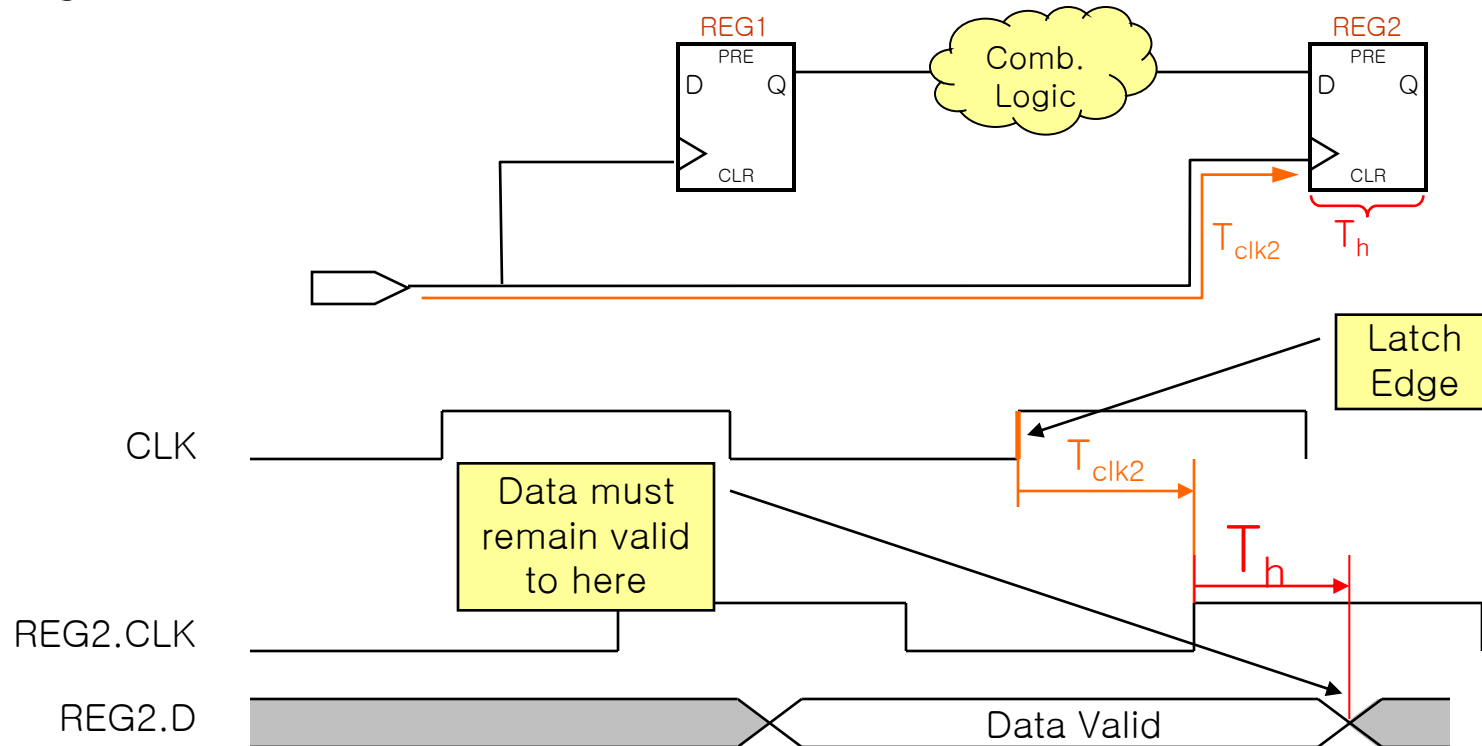
- The minimum time required for the data to get latched into the destination register



$$\text{Data Required Time} = \text{Clock Arrival Time} - T_{su}$$

数据要求时间- Hold

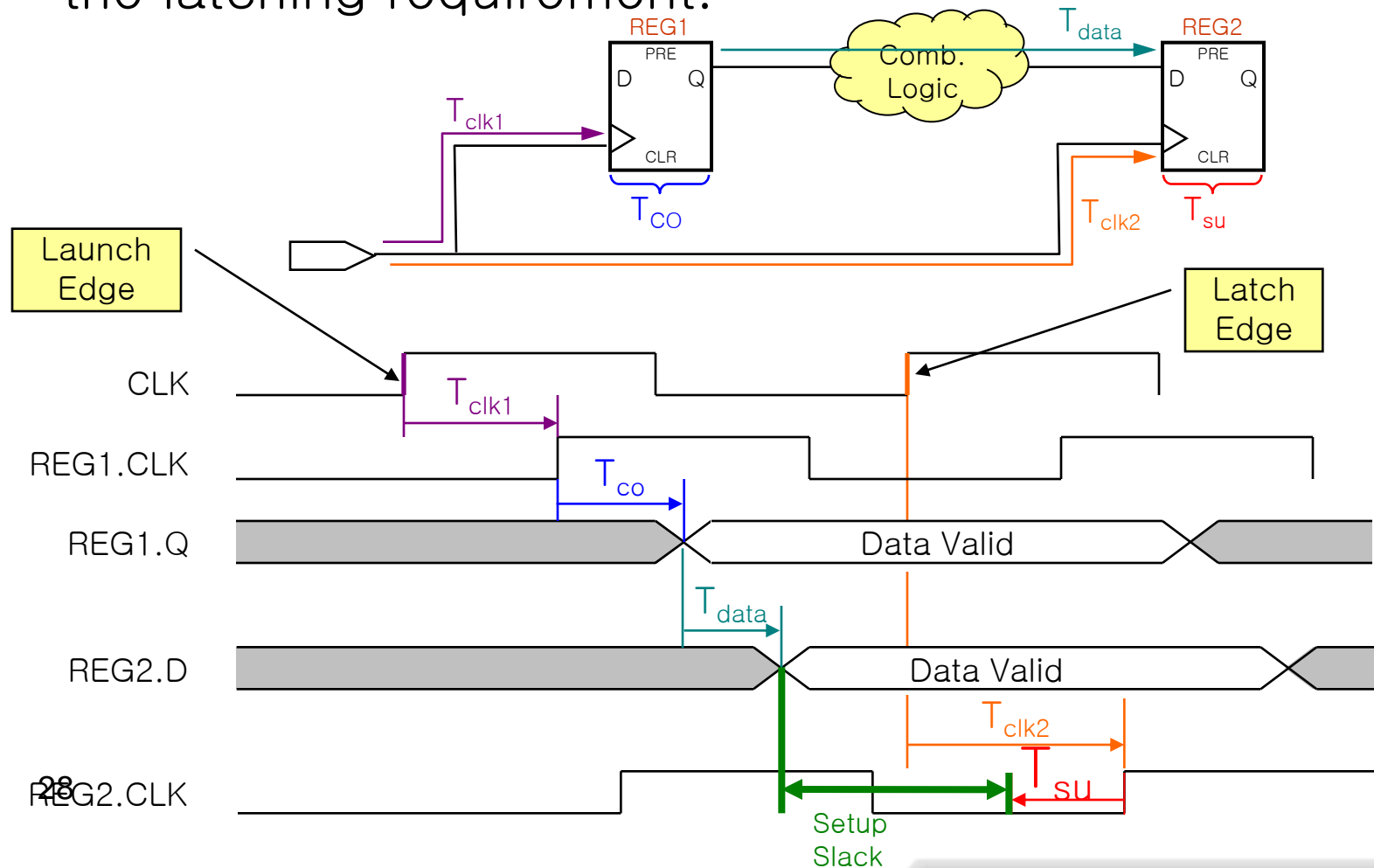
- The minimum time required for the data to get latched into the destination register



$$\text{Data Required Time} = \text{Clock Arrival Time} + T_h$$

Setup Slack

- The margin by which the setup timing requirement is met. It ensures launched data arrives in time to meet the latching requirement.



Setup Slack (cont'd)

$$\text{Setup Slack} = \text{Data Required Time} \\ - \text{Data Arrival Time} \\ (\text{Current})$$

- **Slack**为正，表示数据提前到达，此时组合逻辑延迟满足条件，**Register**有足够的建立时间；
- **Slack**为负，表示数据比预期的时间晚到达，此时显然不满足**Register**的建立时间，不满足约束；

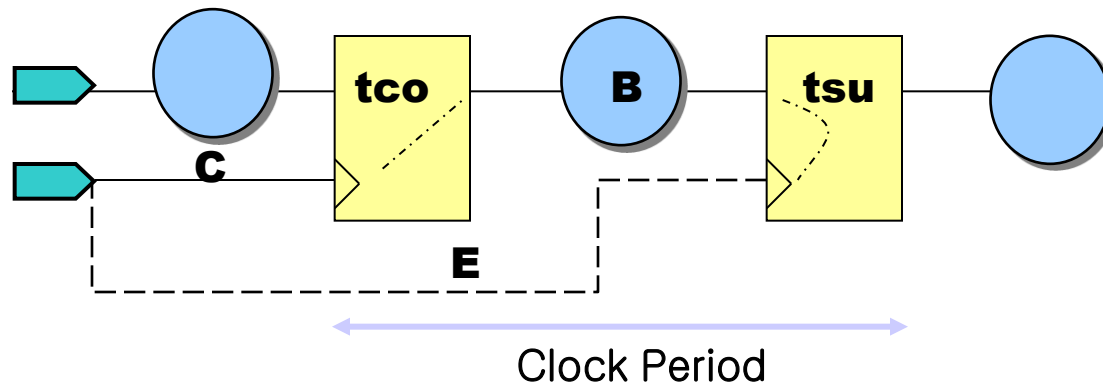
满足建立时间的条件

- 时钟周期应大于寄存器延时、组合逻辑延时、和目标寄存器建立时间的和

$$T > t_{c-q} + t_{logic} + t_{su}$$

Clock Setup (f_{\max})

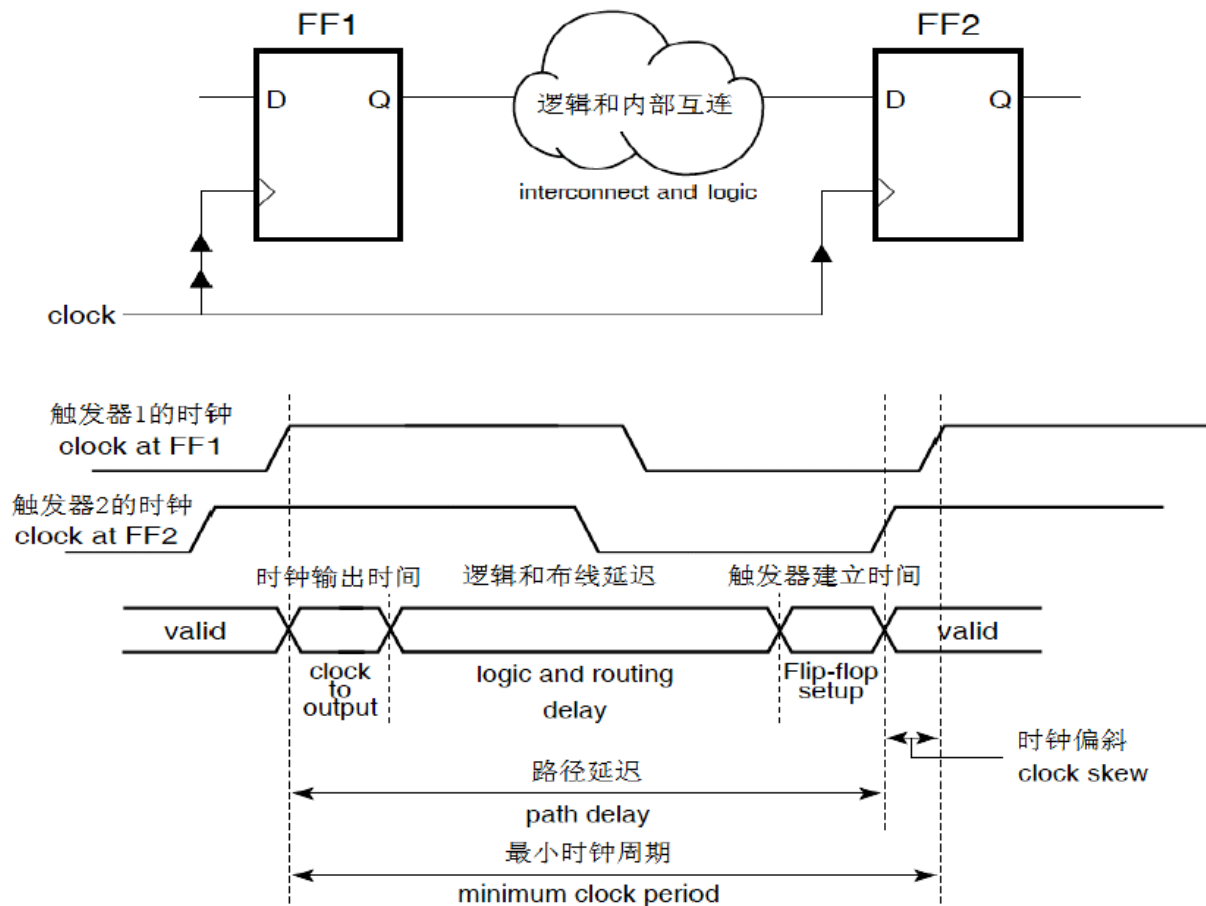
- Worst-Case Clock Frequency



Clock Period = Clock-to-Out + Data Delay + Setup Time - Clock Skew
= $t_{co} + B + t_{su} - (E - C)$

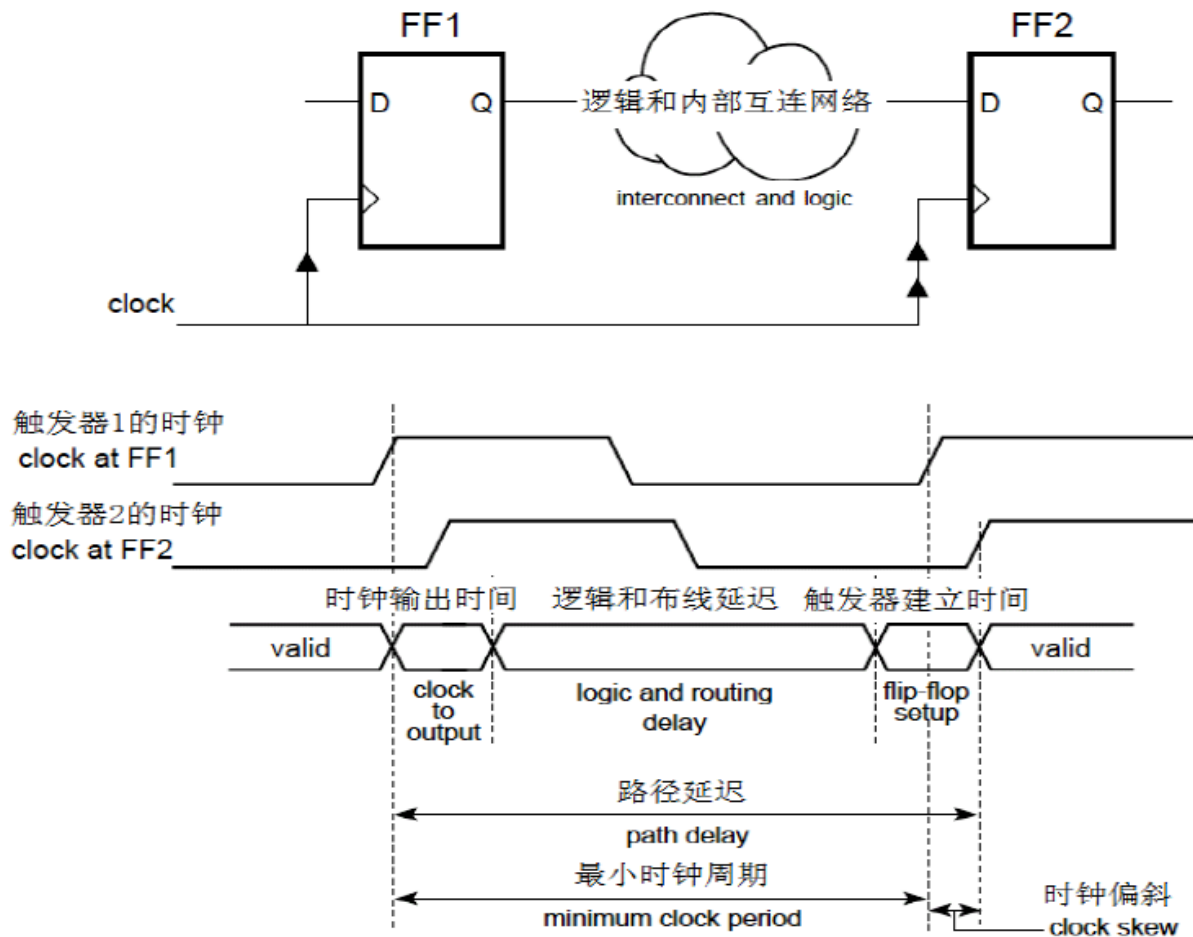
f_{\max} = $1/\text{Clock Period}$

Negative Clock Skew



- 负时钟偏斜情况下最小时钟周期大于数据路径延迟。
- 负时钟偏斜不利于提高工作时钟频率。

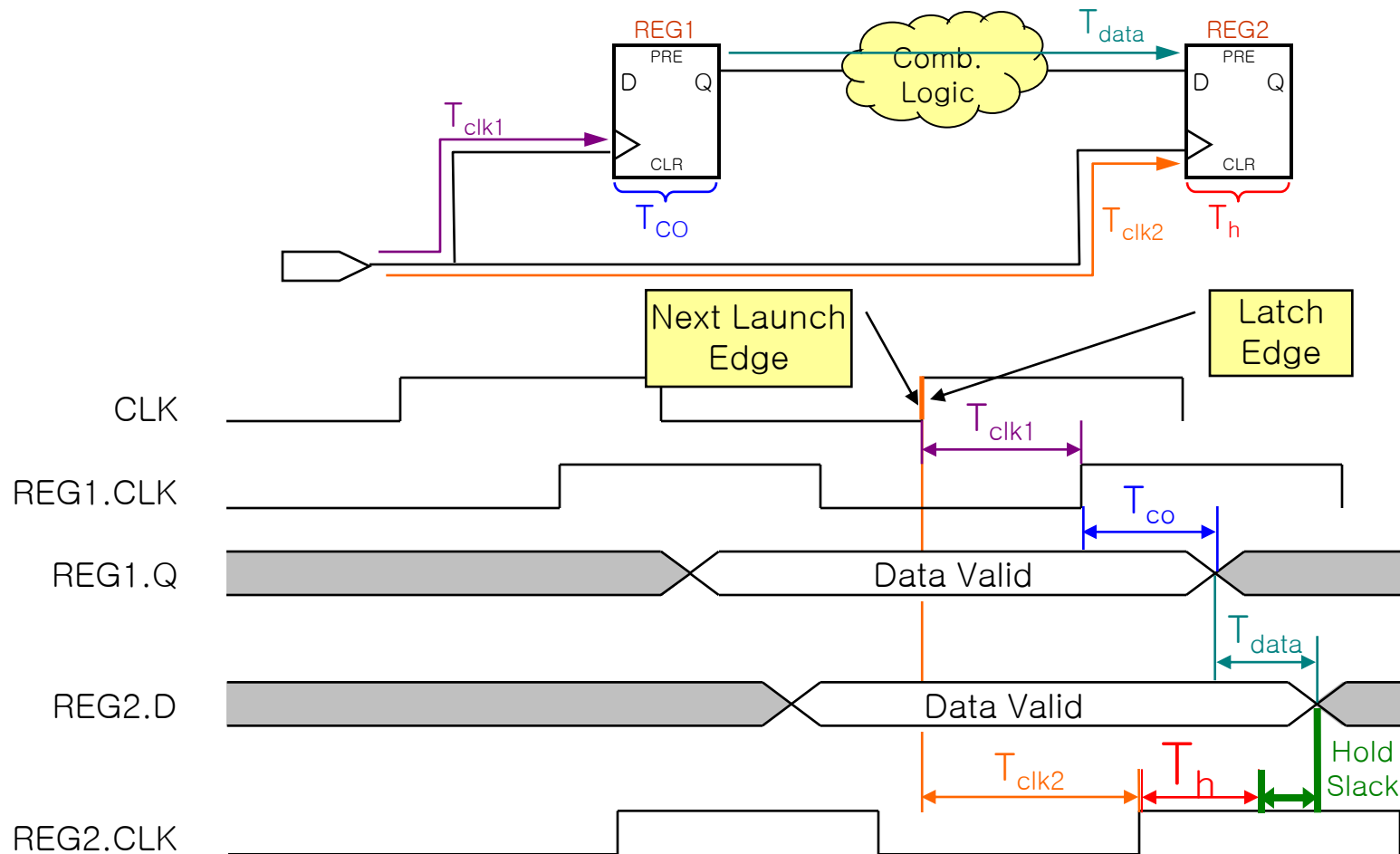
Positive Clock Skew



- 正时钟偏斜情况下，最小时钟周期小于路径延迟，因此有利于提高工作时钟频率。

Hold Slack

- The margin by which the hold timing requirement is met. It ensures latch data is not corrupted by data from another launch edge.



Hold Slack (cont'd)

$$\text{Hold Slack} = \text{Data Arrival Time (Next)} - \text{Data Required Time}$$

Positive slack

- Timing requirement met

Negative slack

- Timing requirement not met

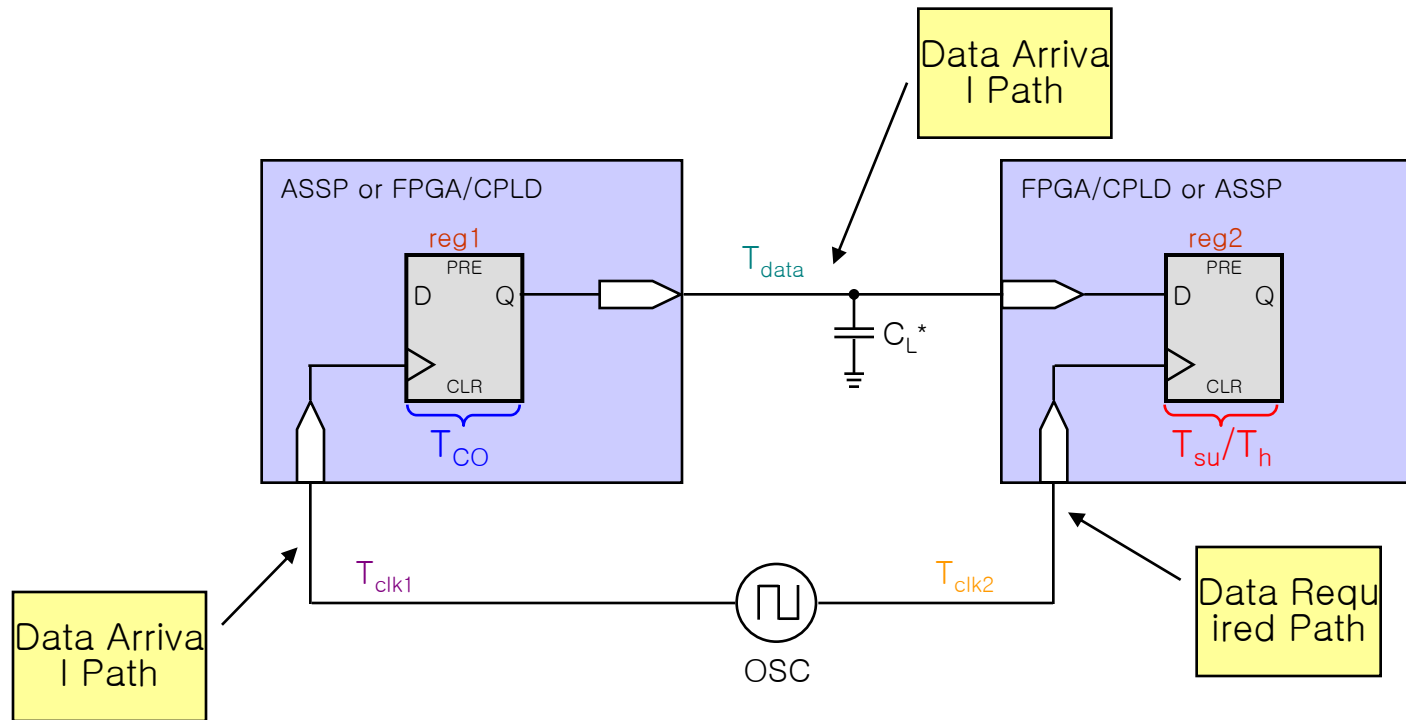
满足保持时间的条件

- 本寄存器有效输出通过组合逻辑的延时应该大于目的寄存器的保持时间要求

$$t_{hold} < t_{c-q, cd} + t_{logic, cd}$$

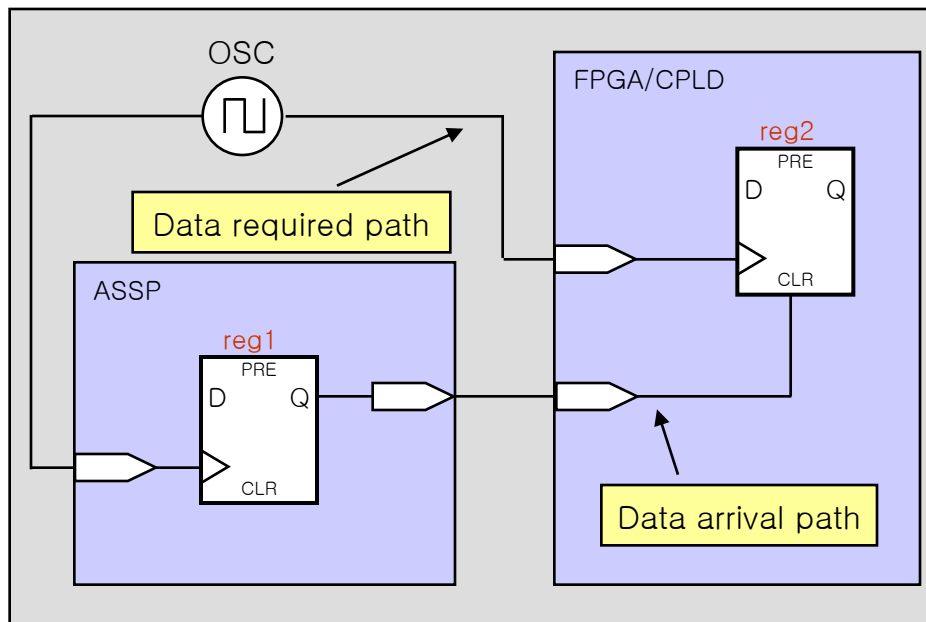
I/O Analysis

- Analyzing I/O performance in a synchronous design uses the same slack equations
 - Must include external device & PCB timing parameters



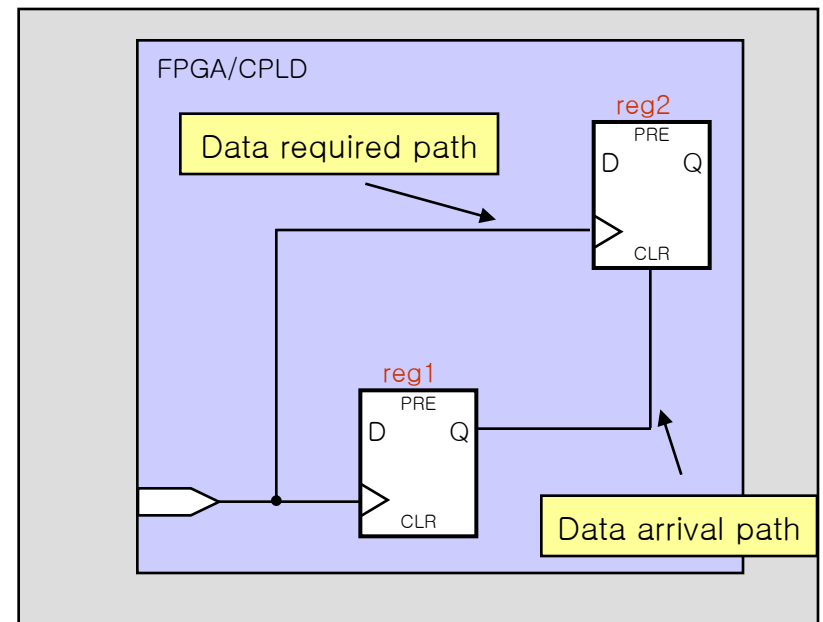
Asynchronous = Synchronous?

- Asynchronous control signal source is assumed synchronous
 - Slack equations still apply
 - data arrival path = asynchronous control path
 - $T_{su} \approx T_{rec}$; $T_h \approx T_{rem}$
 - External device & board timing parameters may be needed



38

Example 1



Example 2

Timing Models

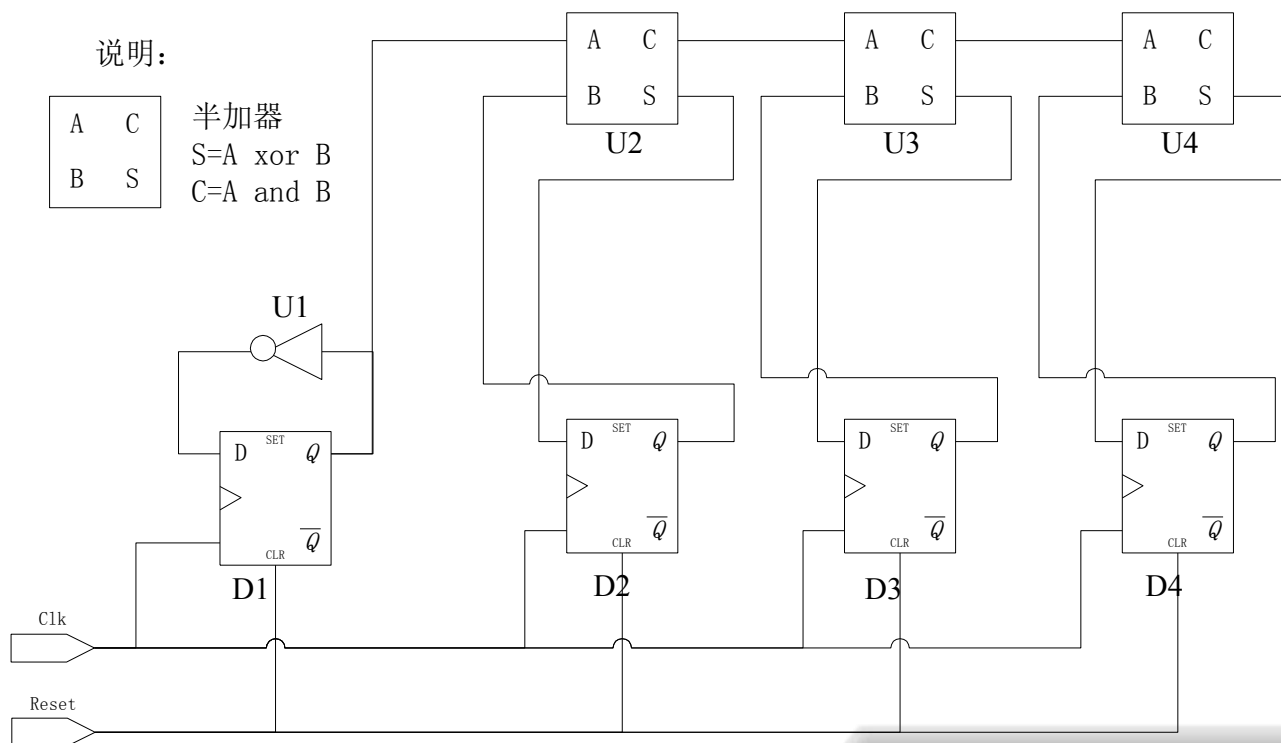
- Software models device timing at two or three PVT conditions by default
 - *Slow Corner* Model
 - Indicates slowest possible performance for any single path
 - Timing for slowest device at maximum operating temperature and VCC_{MIN}
 - *Fast Corner* Model
 - Indicates fastest possible performance for any single path
 - Timing for fastest device at minimum operating temperature and VCC_{MAX}
- Timing closure
 - Ensure both **setup** / **hold** timing is met in **slow** model
 - Ensure **setup** / **hold** timing is met in **fast** model

课程安排

- 静态时序分析基本概念
- 寄存器的基本参数
- 时钟参数满足的条件
- **实例：电路的时序分析方法**

思考题1

- 一个4位的计数器，完成从0x0-0xF的循环计数。它的结构如下图。假设，在正常电压5V时，所有单元门（包括反向器、半加器、选择器以及触发器的时钟到输出端）的延时为1.5ns。触发器的建立时间是1ns，保持时间是1ns。不记线路延时，只考虑门延时。对于时间约束，只考虑系统内部触发器的建立保持时间约束，不考虑其它约束。

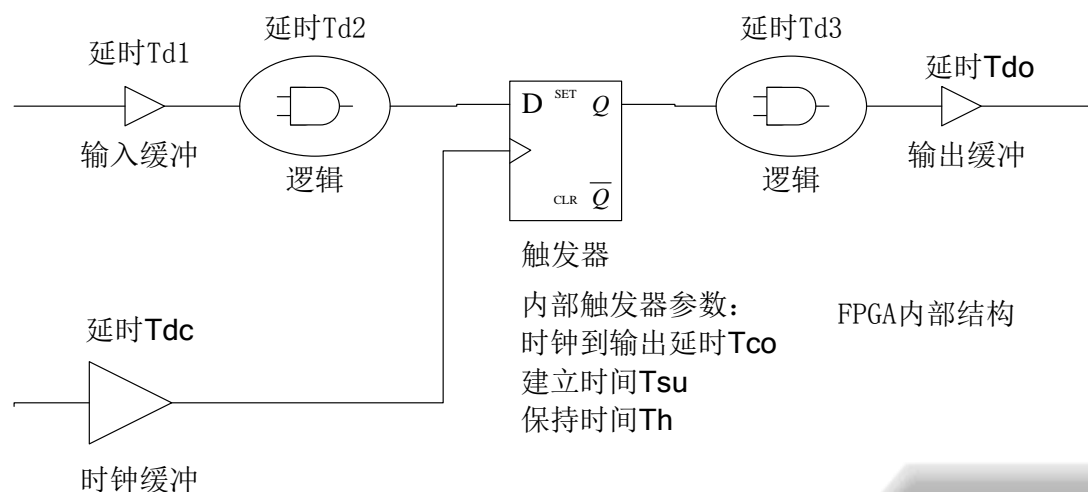


- 问题：1. 请求出在5V电压下，时钟的最高工作频率。
- 2. 假设门延时与电压的关系为：电压每升高0.1V，所有单元门延时减少0.2ns；电压每降低0.1V，延时增加0.2ns，其它参数不变。如果时钟频率是100MHz，求该电路的正常工作电压范围。

思考题2

- 某FPGA的内部结构如下所示。系统中发送的FPGAT通过数据总线Databus驱动K个同样的FPGAR，每个FPGAR都有一定的输入电容 C_R 。数据通过Databus由C1k同步互相传递数据(就是发送数据在第n个时钟，接收数据在第n+1个时钟)。系统时钟C1k的工作频率是50MHz。系统参数(缺省单位为ns)如下

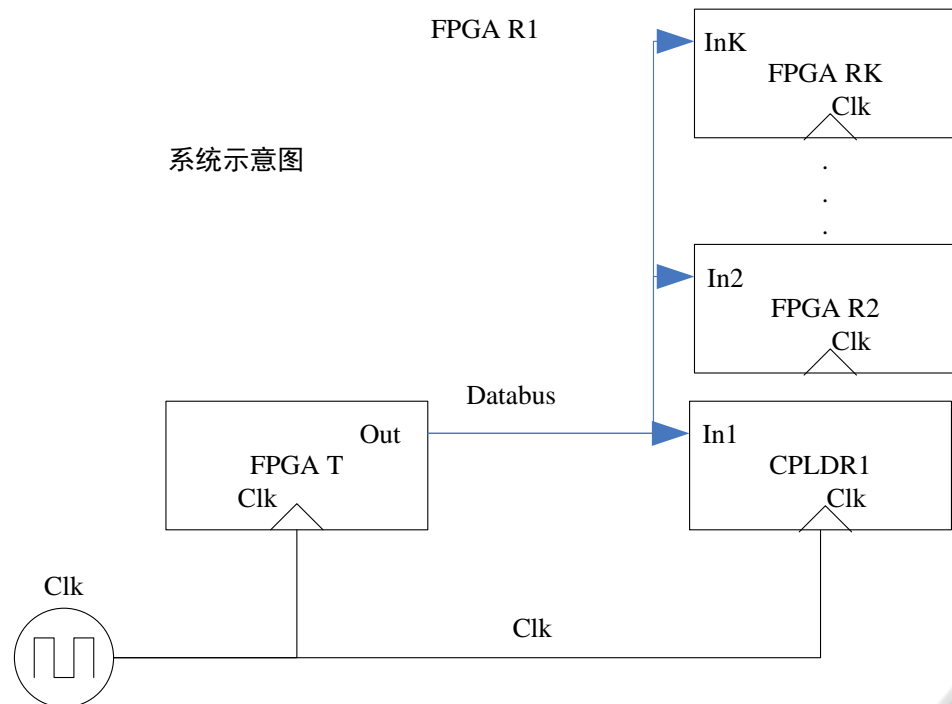
参数	Td1	Td2	Td3	Tdc	Tco	Tsu	Th	Tdo	a	输入电容CR
数值	1	1.5	2.5	2	0.3	0.7	0	1+aCL	0.1ns/pF	15pF



- 输出缓存的延时与外部总负载电容CL有关，为了简化，其延时 T_{do} 近似与CL成线性关系，系数为a。

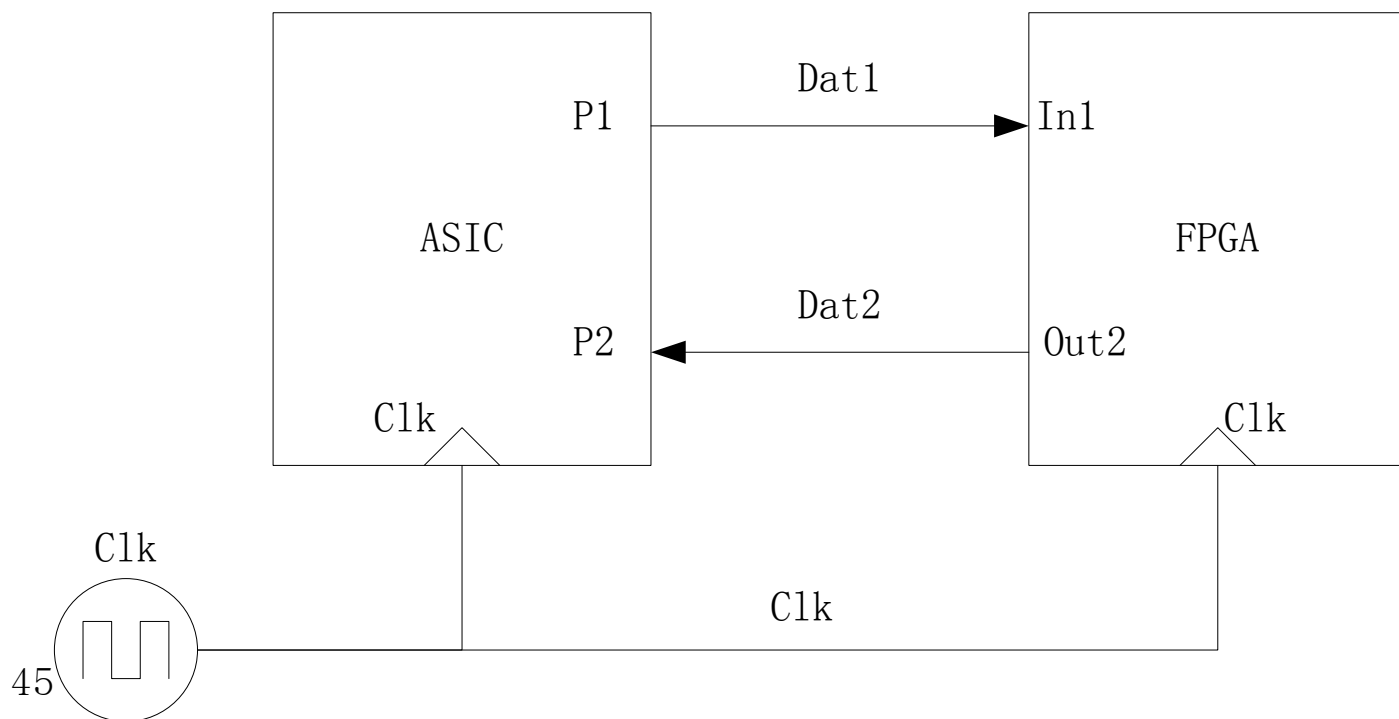
- 问题:

1. 不考虑连线延时，假设Clk是理想时钟，各个FPGA时钟同时到达，那么FPGAT最多可以驱动多少个FPGAR？
2. 当K=2时，如果考虑时钟分配的误差，那么时钟到达FPGAT和FPGAR的最大容许差异是多少？

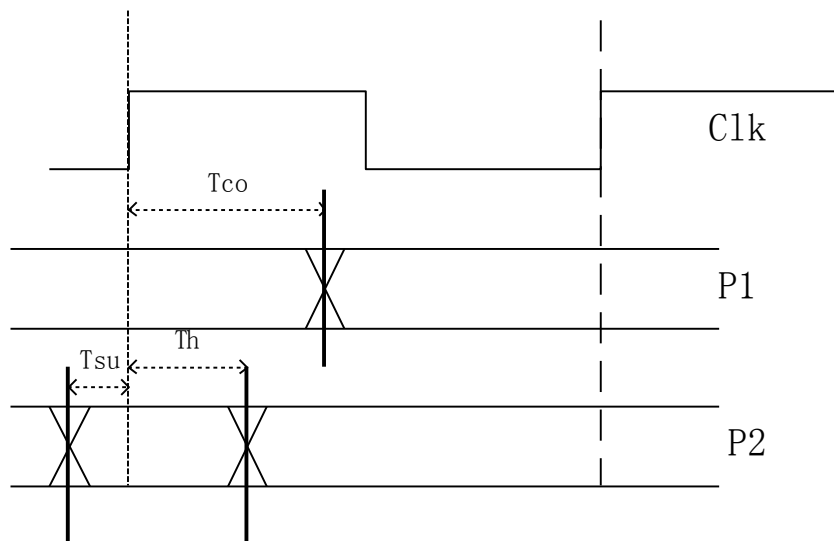


思考题3

- 电路板上包括两个芯片，一个是专用集成电路ASIC，另一个是FPGA。ASIC已经选定。FPGA需要在不同的厂家中选择能够满足技术要求的。电路的结构如下图：



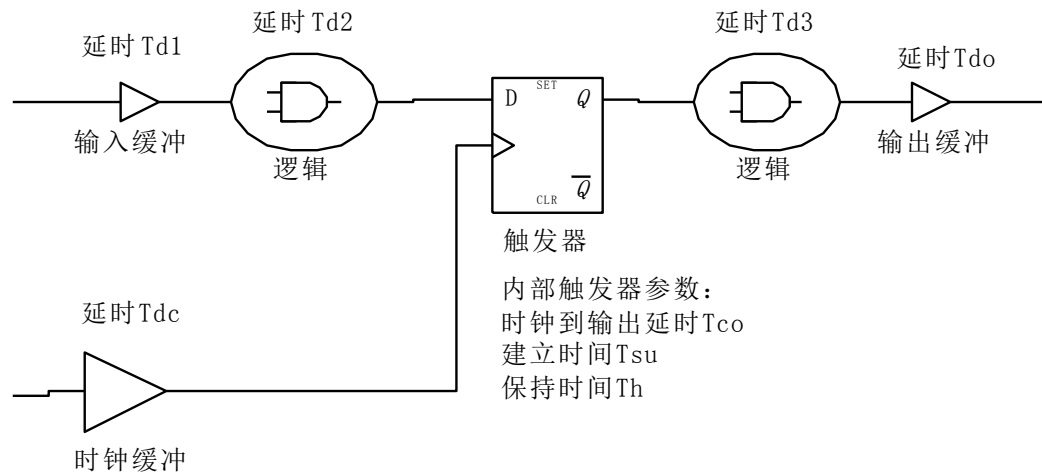
- ASIC与FPGA之间通过Dat1与Dat2按照C1k同步互相传递数据(就是发送数据在第n个时钟, 接收数据在第n+1个时钟)。系统时钟C1k的工作频率是50MHz。ASIC的时间参数如下所示:



46

项目	数值(ns)
P1 从时钟沿到数据有效延时Tco1	8
P2 的建立时间Tsu1	6
P2 的保持时间Th1	3

- FPGA的内部结构如下：



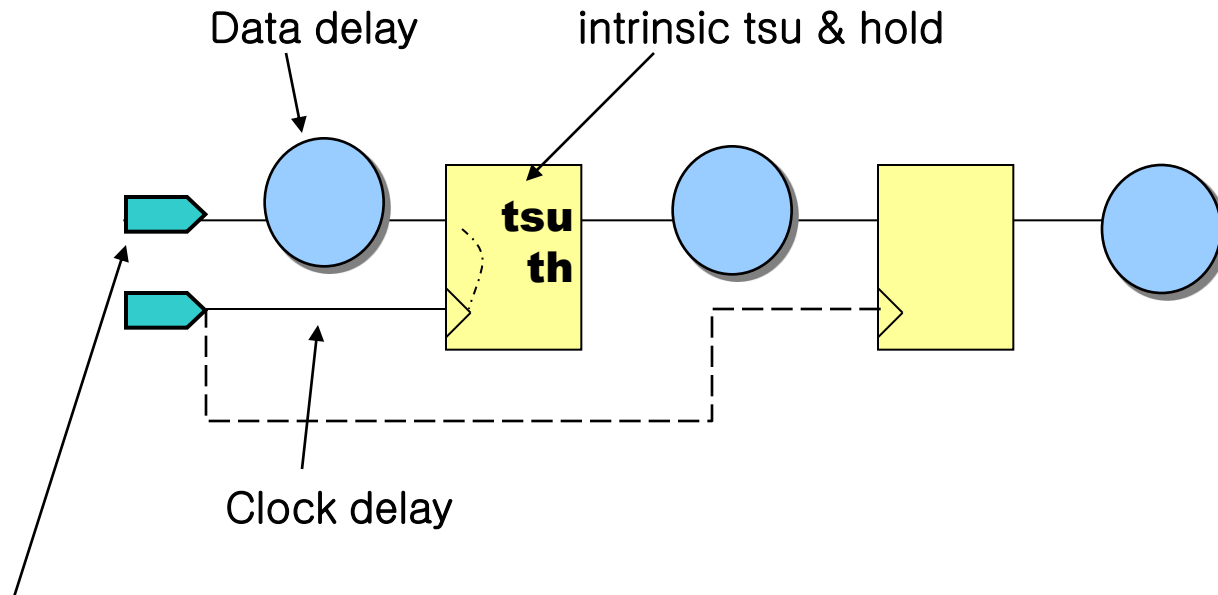
三个厂家的参数分别如下,单位ns：

厂家	T_{d1}	T_{d2}	T_{d3}	T_{do}	T_{dc}	T_{co}	T_{su}	T_h
A	1	7	7	1	3	1	2	0
B	0.3	1	1	0.3	1	0.5	1	0.1
C	2	10	10	2	3	5	6	2

1. 请问哪些厂家的FPGA合适？

47 2. 只对ASIC与FPGA互相正确传递数据而言，合格厂家FPGA应用系统的最高工作频率是多少？

I/O Setup (tsu) & Hold (th) Analyses

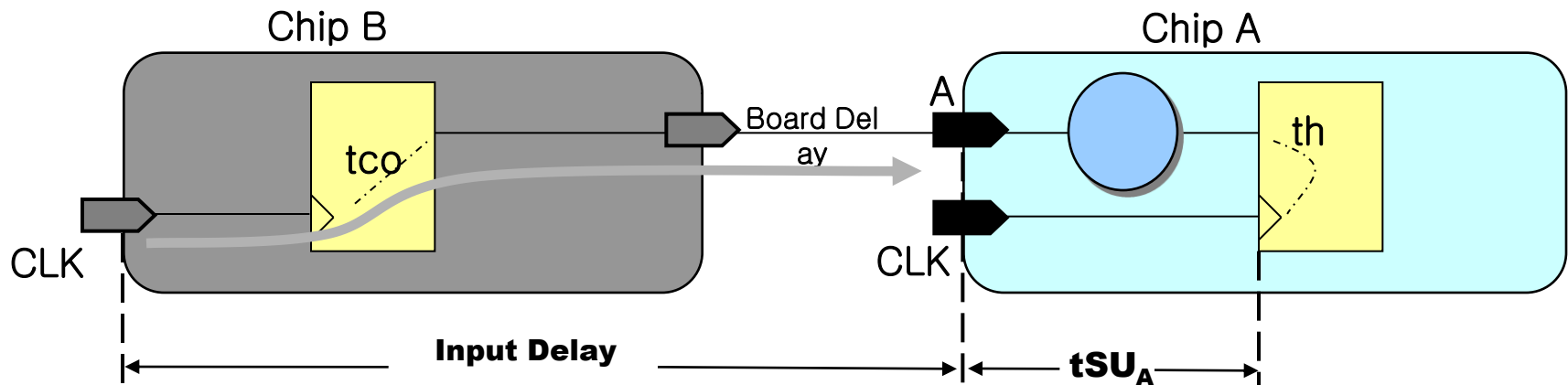


$$tsu = \text{data delay} - \text{clock delay} + \text{intrinsic tsu}$$

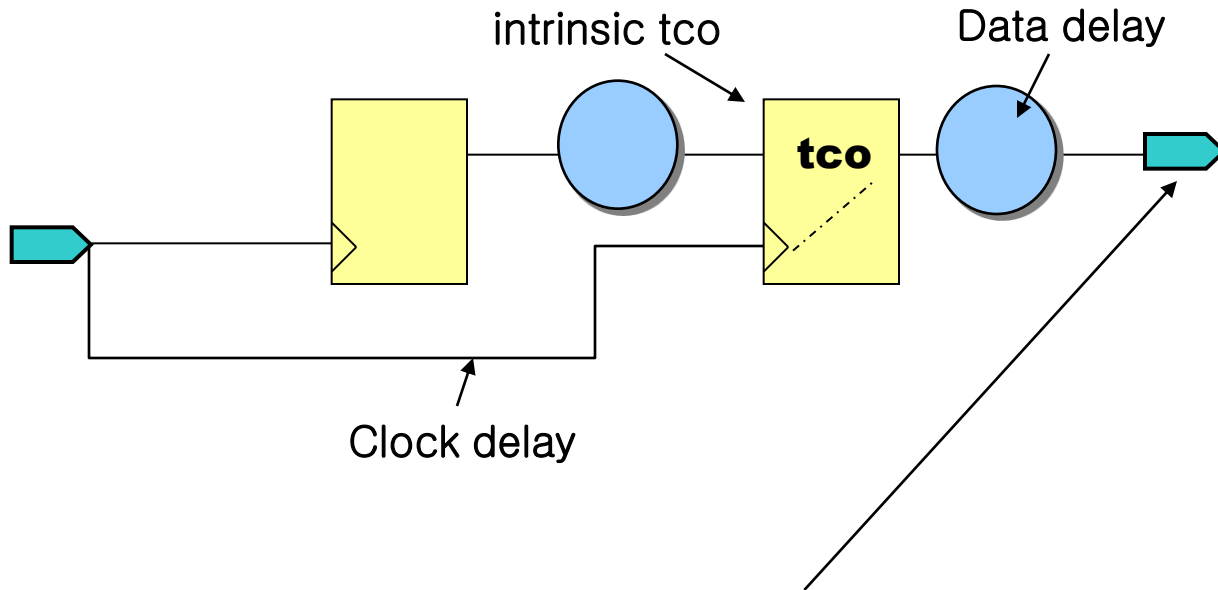
$$th = \text{clock delay} - \text{data delay} + \text{intrinsic th}$$

例：同时钟域的两个器件间的tsu/th时序关系

- $t_{suA} \leq T_{CLK} - t_{coB} - \text{board delay}$
- $t_{thA} \leq t_{coB} + \text{board delay}$



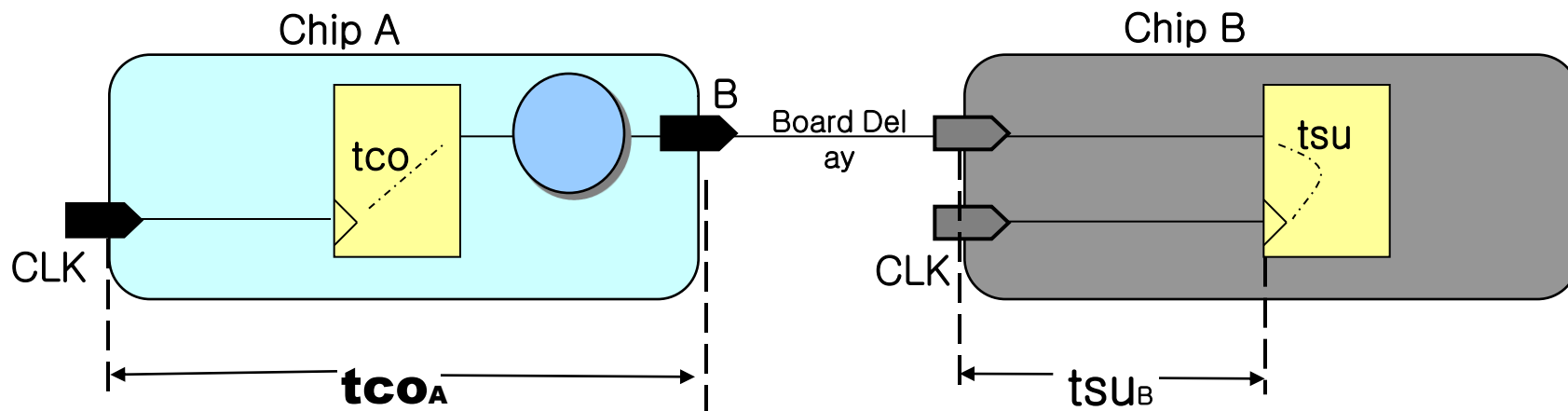
I/O Clock-to-Output Analysis (tco)



$$\text{clock delay} + \text{intrinsic tco} + \text{data delay} = \text{tco}$$

例：同时钟域的两个器件间的tco时序关系

- $tco_A \leq T_{clk} - \text{Board Delay} - tsu_B$



总结

- 静态时序分析在数字同步电路设计中有着举足轻重的作用。
- 建立时间、保持时间的满足条件需要牢牢掌握，这是硬件设计的基础保证。
- 有了前面的基础，才能正确进行时序约束，设计出良好的高频工作电路。