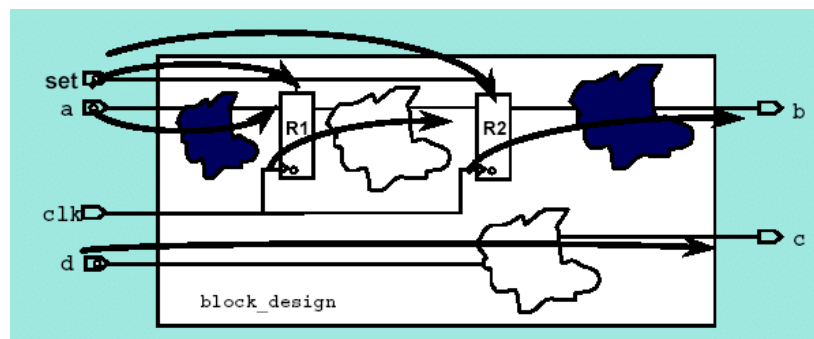


## BTS硬件逻辑培训体系（2）

# 静态时序分析与逻辑设计



华为上研硬件BTS

2003年4月

Version 1.0

# 课程内容介绍

---

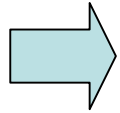
静态时序分析是一种重要的逻辑验证方法，设计者根据静态时序分析的结果来修改和优化逻辑，直到设计满足要求。

课程内容：

- ◆ 静态时序的概念、目的（**要求掌握**）
- ◆ 静态时序分析路径（要求了解）
- ◆ 静态时序分析方法（**要求掌握**）
- ◆ 静态时序分析工具介绍（**要求掌握**Timing Analyzer）
- ◆ 静态时序分析报告及逻辑设计优化（要求了解）

# 课程目录

---



- ◆ 静态时序分析—概念与流程
- ◆ 静态时序分析—时序路径
- ◆ 静态时序分析—分析工具

## 静态时序分析-Static Timing Analysis (STA)

---

- 静态时序分析是一种验证方法
- 静态时序分析的前提是同步逻辑设计。
- 静态时序分析工具通过路径计算延迟的总和，并比较相对于预定义时钟的延迟
- 静态时序分析仅关注时序间的相对关系而不是评估逻辑功能
- 无需用向量去激活某个路径，对所有的时序路径进行错误分析，能处理百万门级的设计，分析速度比时序仿真工具快几个数量级。在同步逻辑情况下，可以达到**100%**的时序路径覆盖
- 静态时序分析的目的是找出隐藏的时序问题，根据时序分析结果优化逻辑或约束条件，使设计达到时序闭合（**timing closure**）

# 静态时序分析-Static Timing Analysis (STA)

---

- 静态时序工具可识别的时序故障数要比仿真多得多，包括：建立/保持和恢复/移除检查(包括反向建立/保持)；最小和最大跳变；时钟脉冲宽度和时钟畸变；门级时钟的瞬时脉冲检测；总线竞争与总线悬浮错误；不受约束的逻辑通道
- 一些静态时序工具还能计算经过导通晶体管、传输门和双向锁存的延迟，并能自动对关键路径、约束性冲突、异步时钟域和某些瓶颈逻辑进行识别与分类

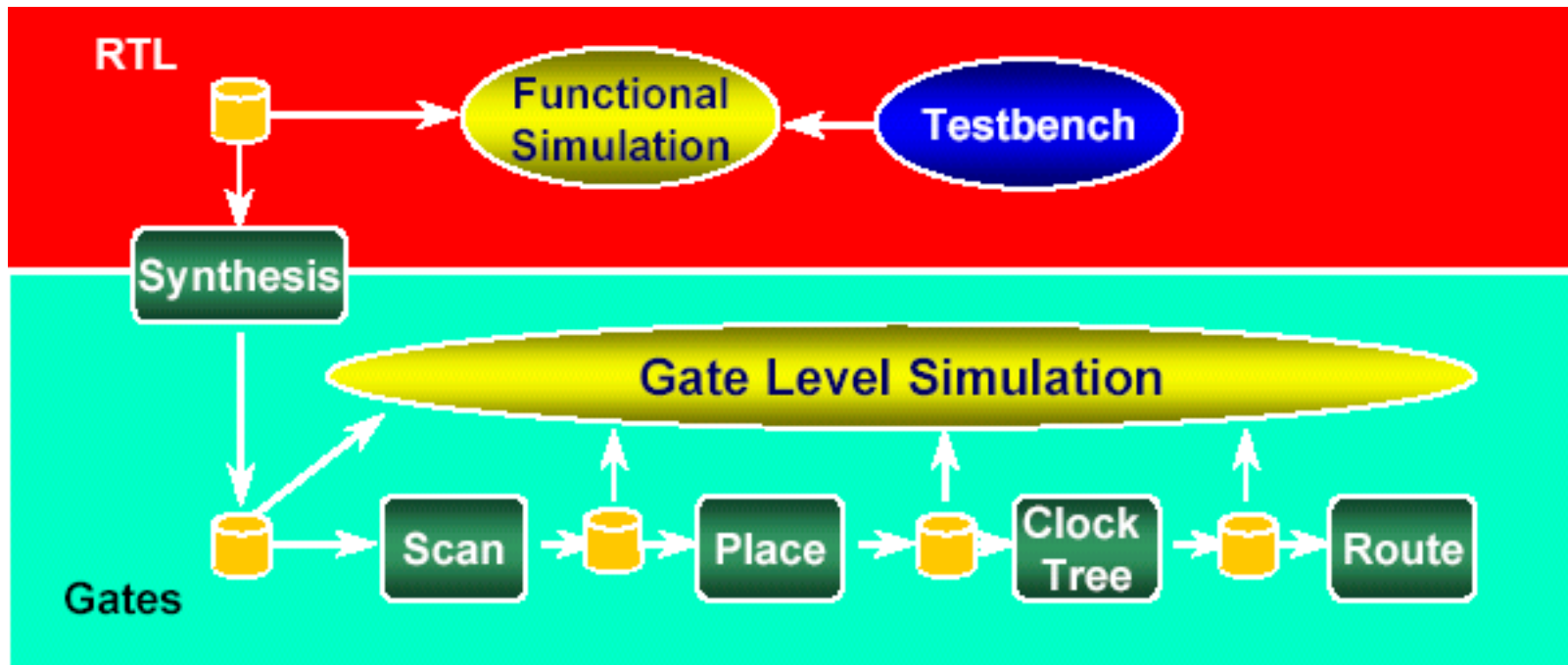
## 静态时序分析-Static Timing Analysis (STA)

---

- ◆ Synopsys公司的 **PrimeTime**和Mentor Graphics公司的SST Velocity主要用于全芯片的IC设计。Primetime是业界最流行的分析工具。
- ◆ Innoveda公司的Blast工具主要用于FPGA和板级设计，它可支持单层和多层板结构，能链接板级交叉耦合分析。
- ◆ 各FPGA厂商的工具均提供静态时序分析功能。FPGA的静态时序分析比IC简单。

# 早期的ASIC验证流程

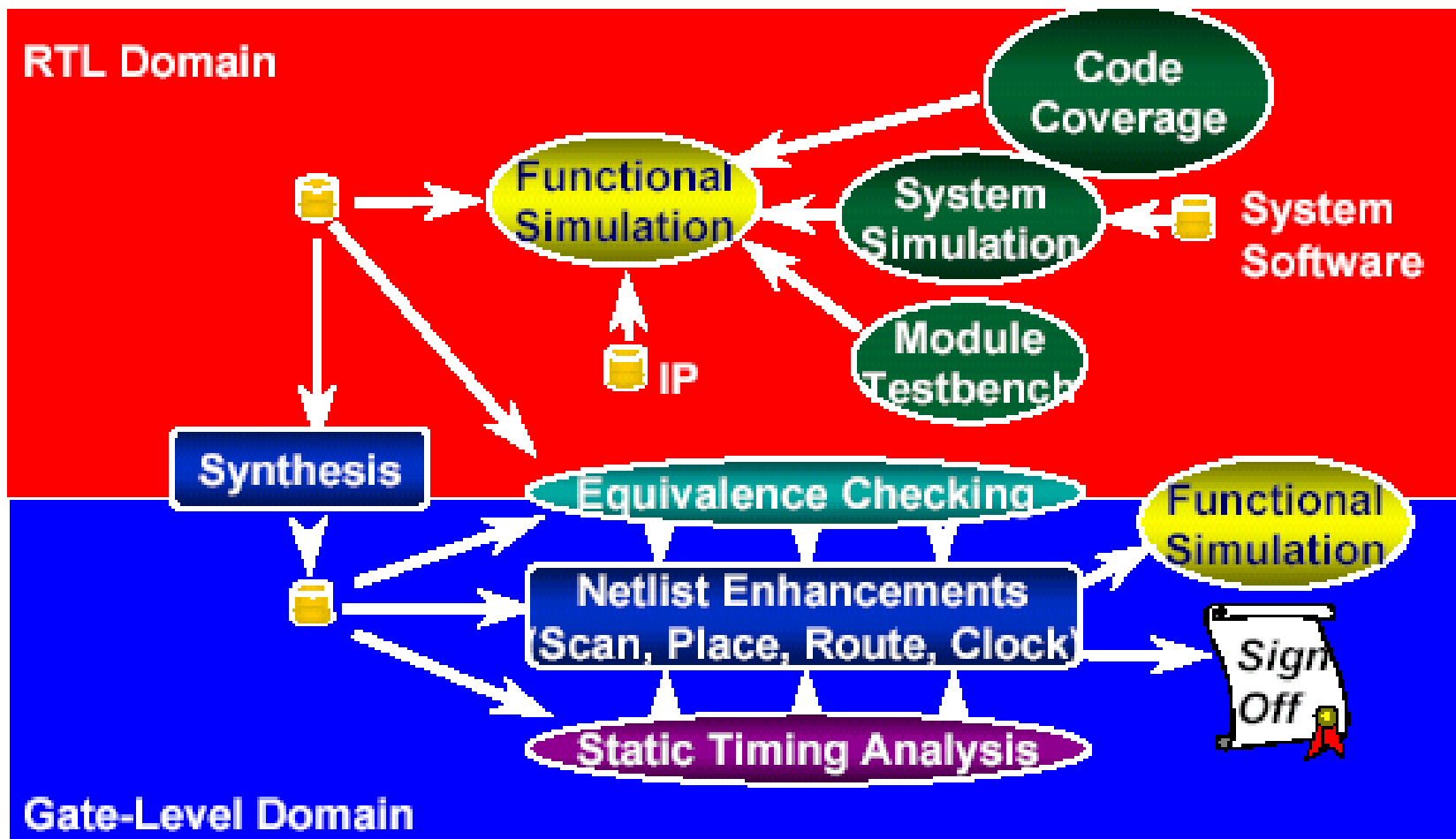
## 早期的ASIC验证流程



- ◆ 需要额外的仿真和测试向量
- ◆ 适用于小规模FPGA和小规模ASIC
- ◆ 不适用于大规模逻辑（百万门级）

# 目前的ASIC验证流程

目前的ASIC验证流程

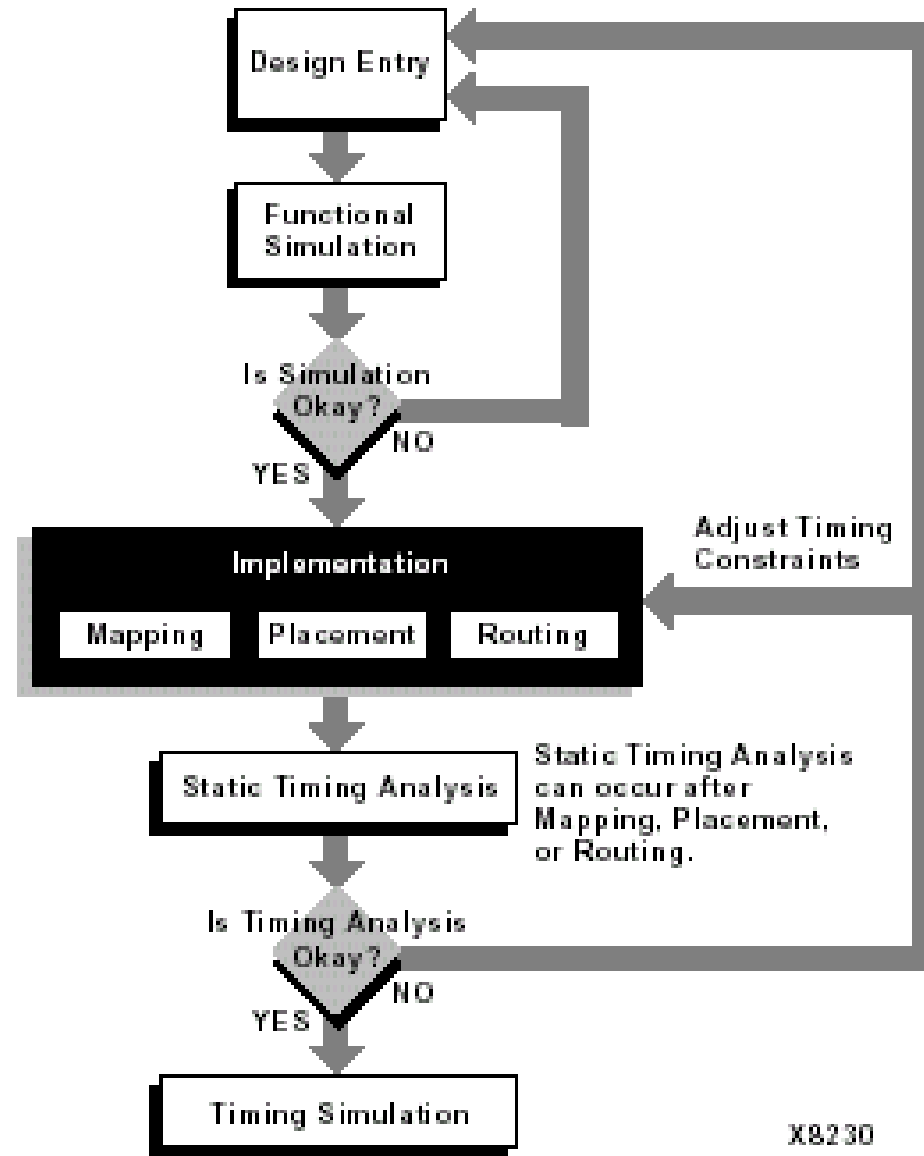




# FPGA逻辑设计与验证流程

## FPGA逻辑设计与验证流程

FPGA验证流程是简化的ASIC验证流程



# 逻辑设计与逻辑验证

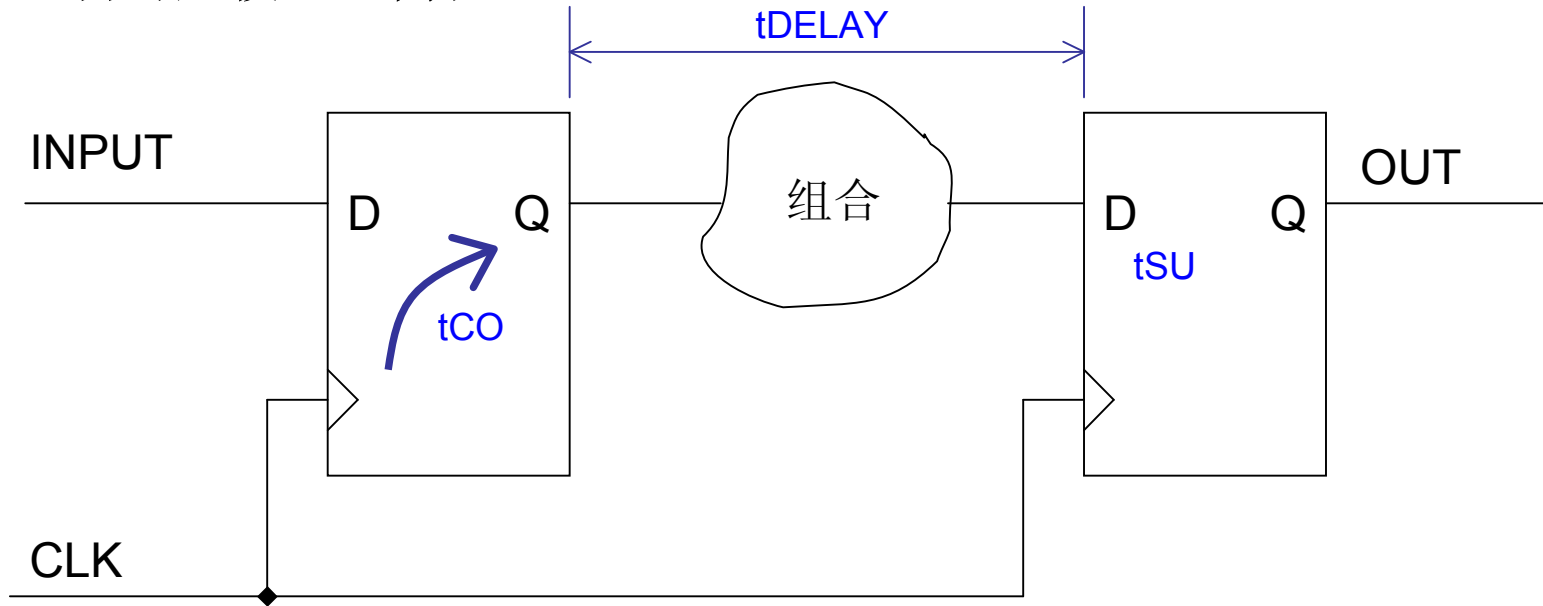
---

## 逻辑设计与逻辑验证

- ◆ 逻辑设计完成后必须进行逻辑验证
- ◆ 逻辑验证是为了修改和优化逻辑设计
- ◆ 动态仿真、静态时序分析、形式验证是常用的验证方法
- ◆ 静态时序分析工具可以输出分析报告，根据分析报告进行优化逻辑设计，直到设计满足要求

# 同步逻辑时延模型

同步逻辑时延模型（简化）



$$T = t_{CO} + t_{DELAY} + t_{SU}$$

时钟周期大于T，触发器正常工作；

时钟周期小于T，不满足建立时间，触发器可能经历亚稳态。

最高时钟频率为  $f = 1/T$

# 建立/保持时间(Setup/Hold time)

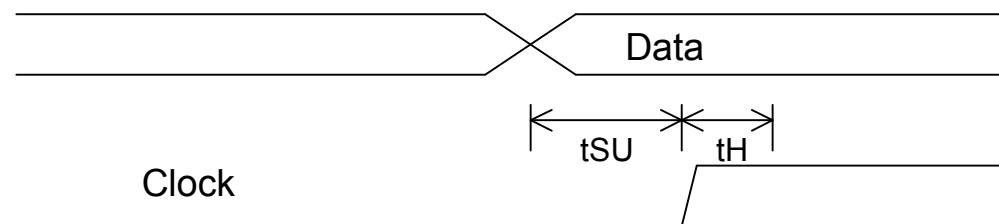
---

## 建立时间 (Setup time) $t_{SU}$

触发器的时钟沿到来之前，输入数据稳定不变的时间。

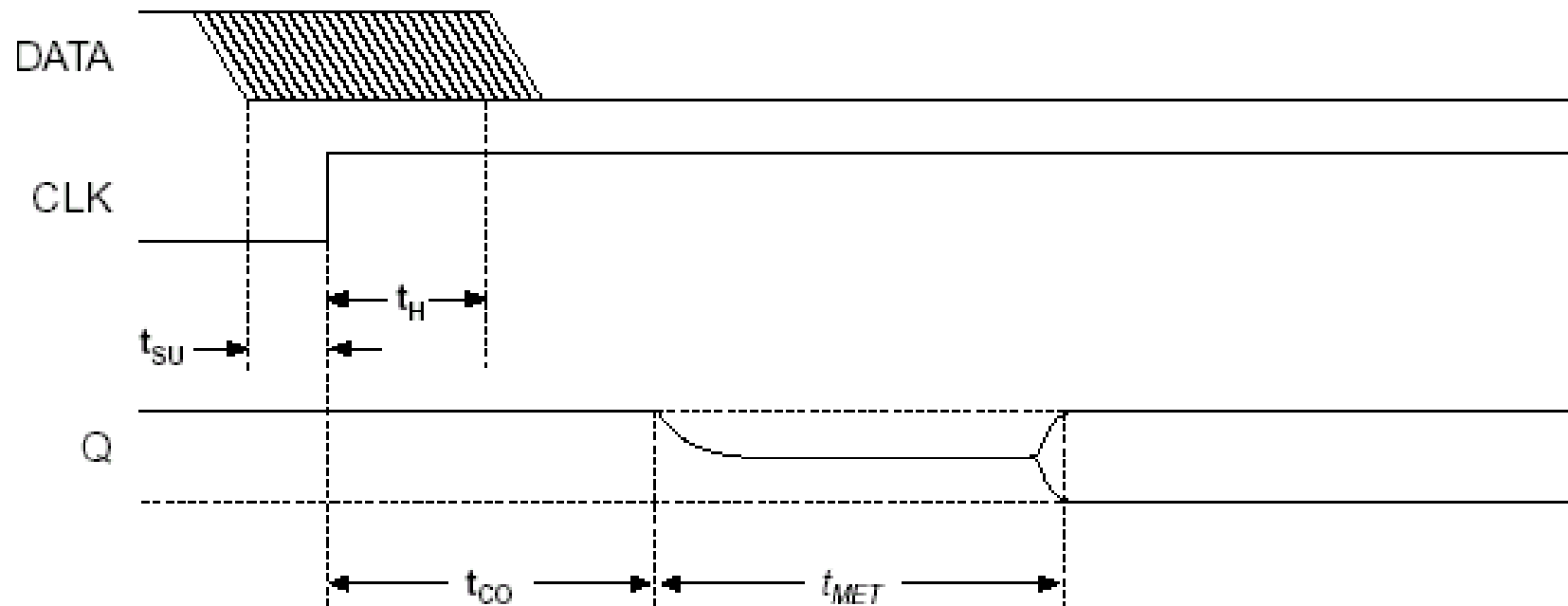
## 保持时间 (Hold time) $t_H$

触发器的时钟沿到来之后，输入数据稳定不变的时间。



# 建立/保持时间(Setup/Hold Time)

不满足建立/保持时间时，触发器可能会经历亚稳态



$t_{SU}$  - 建立时间 (Setup time) ,  $t_H$  - 保持时间 (Hold time)

$t_{CO}$  - 时钟到输出时间 (Clock-to-Output time)

$t_{MET}$  - 建立时间 (Setting time) (亚稳态到稳态的时间, 与器件工艺有关)

# 时序约束(Timing Constraints)

---

时序约束，是指在逻辑综合、布局布线或静态时序分析时，在综合工具、布局布线（P&R）工具或静态时序分析（STA）工具中指定信号的频率/周期、占空比、时延等约束条件，EDA工具根据设定的约束条件来工作。

- ◆ 在综合工具中指定时序约束条件，目标是使综合生成的电路满足时序约束要求。
- ◆ 在布局布线工具中指定约束条件，目标是使布局布线的结果满足时序约束要求。
- ◆ 在静态时序分析工具中指定约束条件，目标是根据约束条件来分析设计是否满足要求。

# 时序闭合(Timing Closure)

---

时序闭合，又称时序收敛。是指通过在逻辑综合工具、布局布线工具中指定时序约束条件以进行综合和布局布线，然后根据静态时序分析的结果，经过优化设计或修改约束条件后，使设计满足时序约束条件。

简单地说，时序闭合就是指设计满足时序约束条件。

目前IC设计工具不断发展的方向是，将逻辑综合、布局布线、静态时序分析紧密结合在一起，使设计一次性达到时序闭合为最高目标。

## 关键路径(Critical Path)

---

关键路径通常是指同步逻辑电路中，组合逻辑时延最大的路径。也就是说关键路径是对设计性能起决定性影响的时序路径。

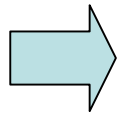
静态时序分析能够找出逻辑电路的关键路径。通过查看静态时序分析报告，可以确定关键路径。

对关键路径进行时序优化，可以直接提高设计性能。对同步逻辑来说，常用的时序优化方法包括Pipeline、Retiming两种方法。



# 课程目录

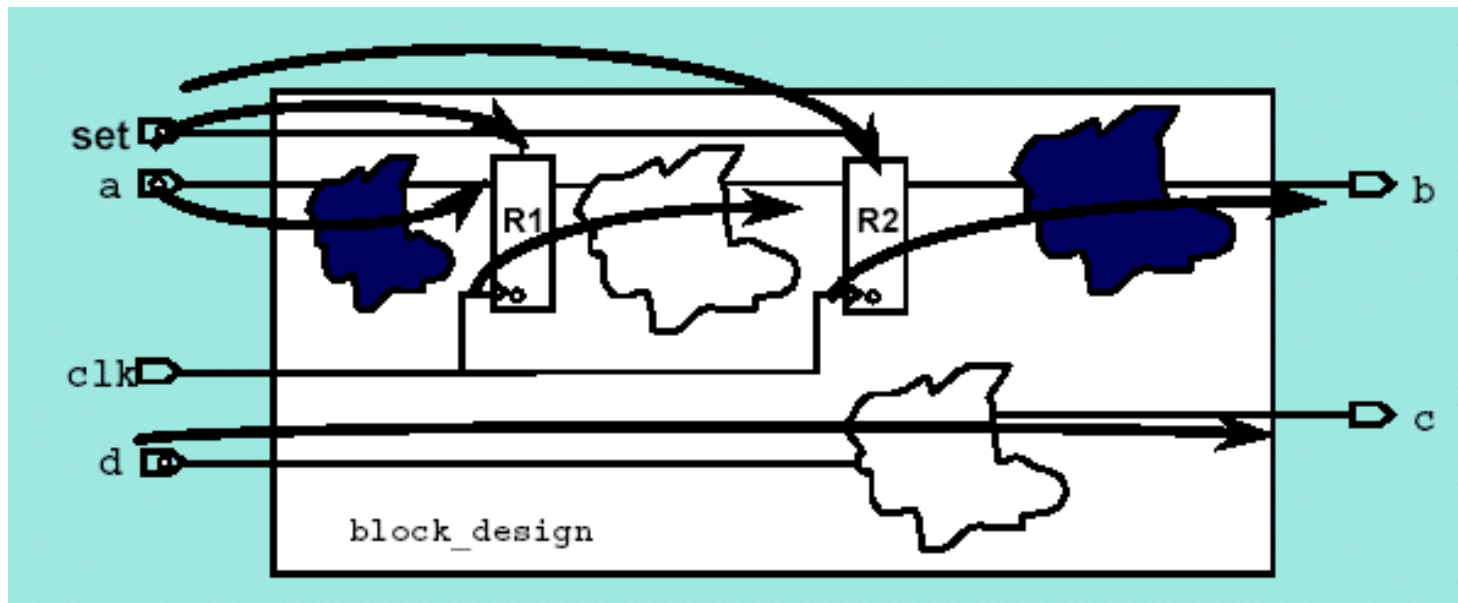
---



- ◆ 静态时序分析—概念与流程
- ◆ 静态时序分析—时序路径
- ◆ 静态时序分析—分析工具

# 时序路径(Timing Paths)

## 典型时序路径



① 输入端口到触发器

② 触发器到触发器

③ 触发器到输出端口

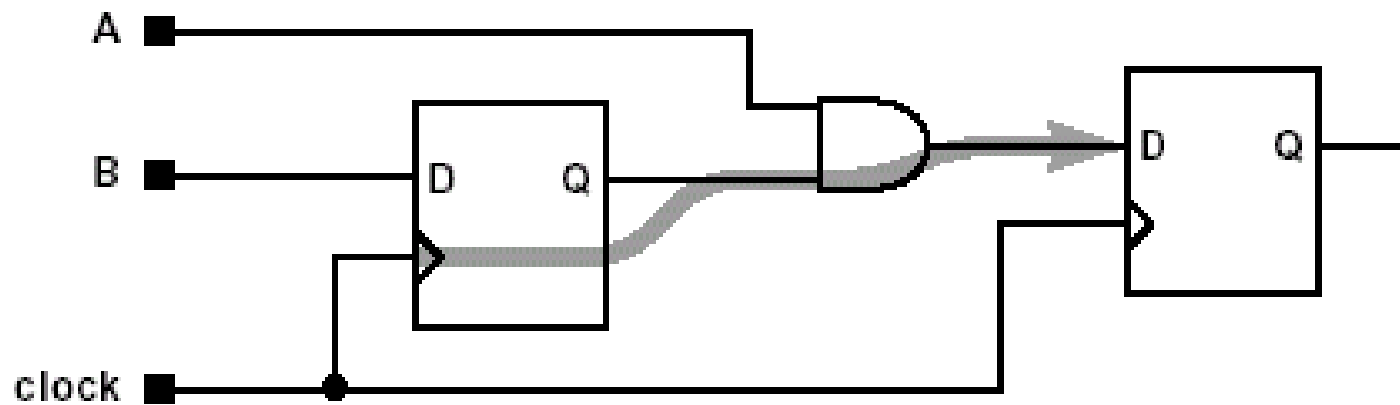
④ 输入端口到输出端口

# 时钟到建立(Clock-to-Setup Path)

---

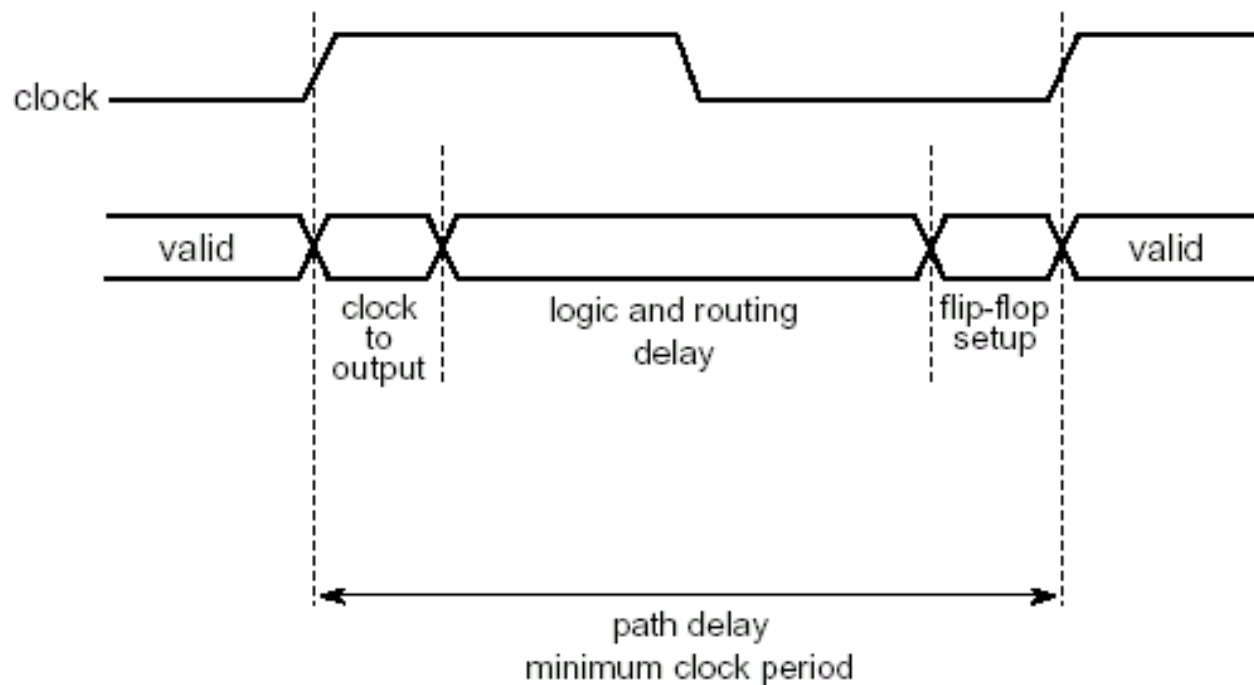
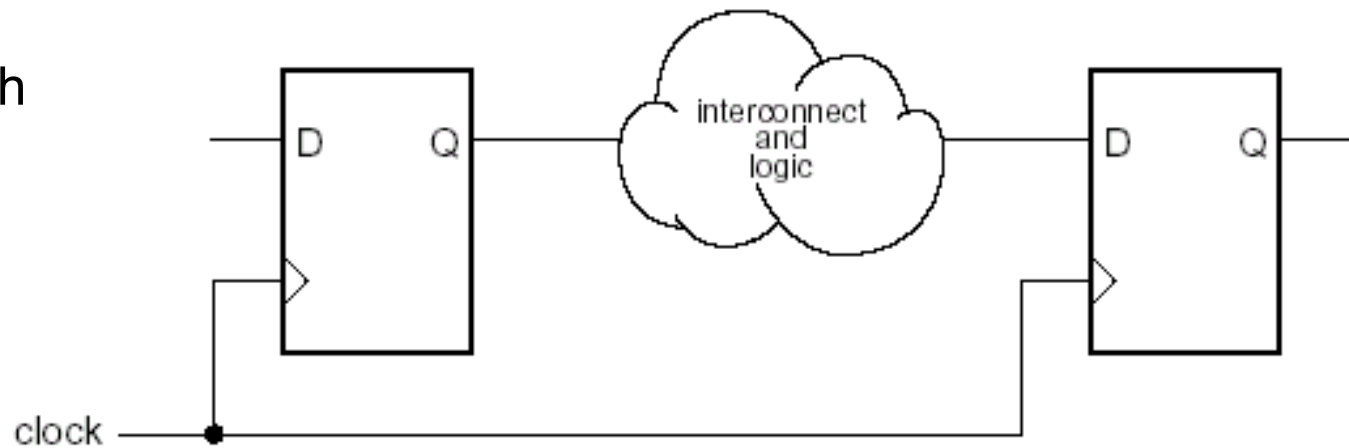
Clock-to-Setup Path

tSU - 时钟到建立时延



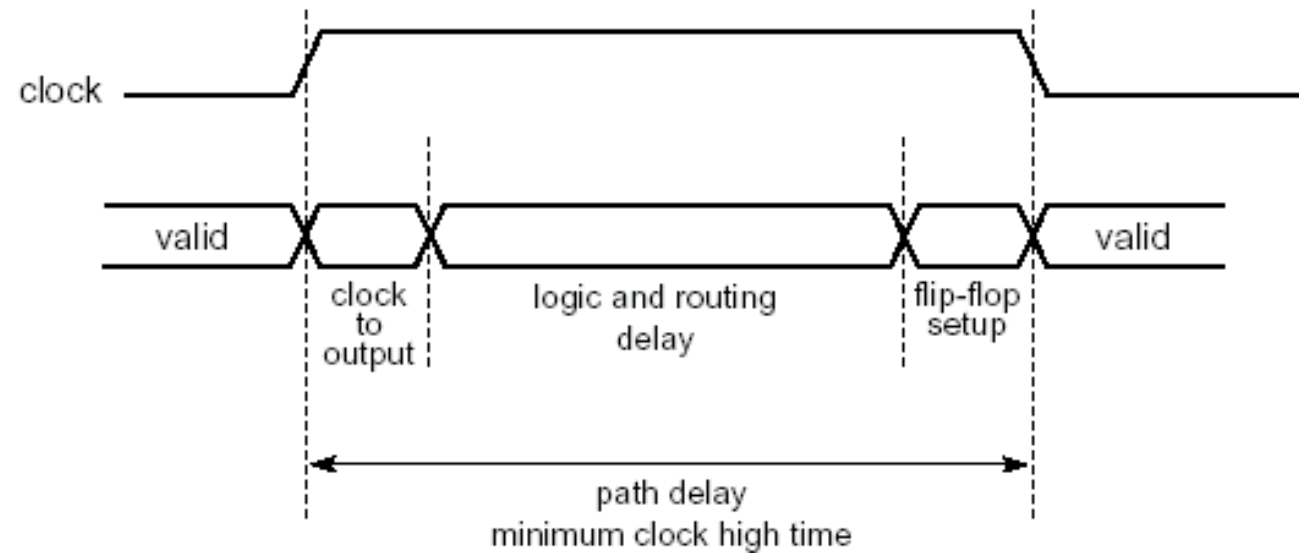
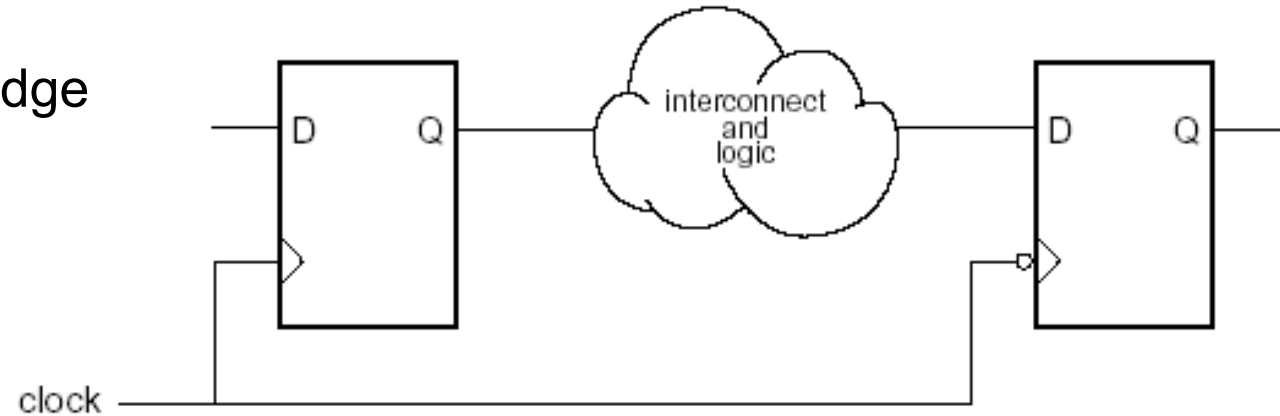
# 时钟到建立(Clock-to-Setup Path)

Clock-to-Setup Path



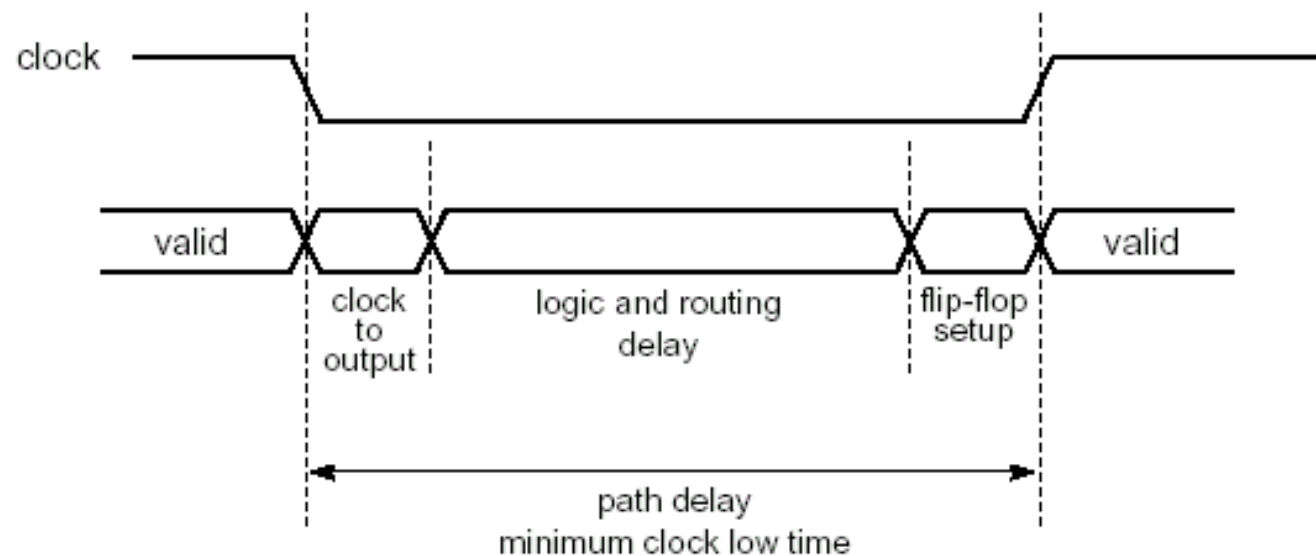
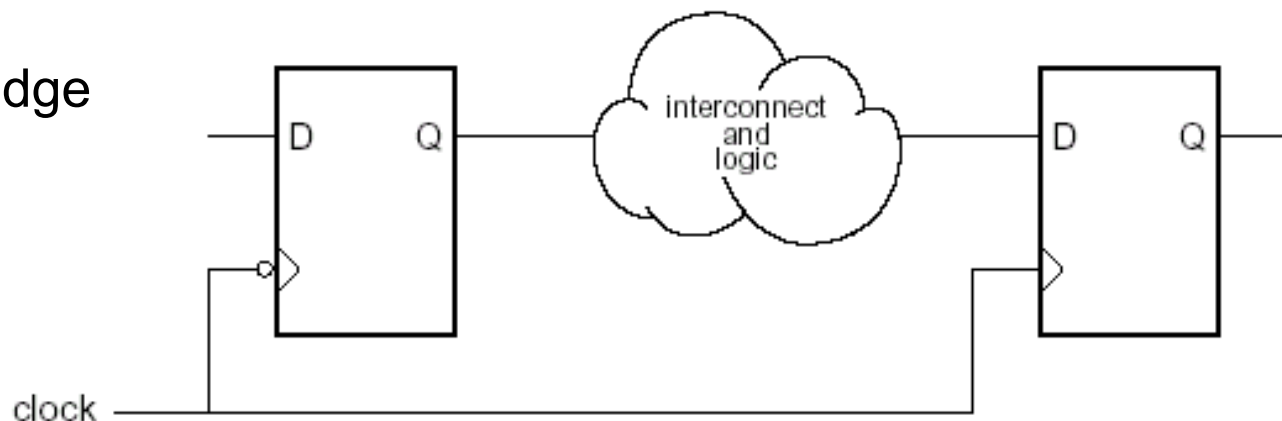
# 时钟到建立(Clock-to-Setup Path)

Rising to Falling Edge



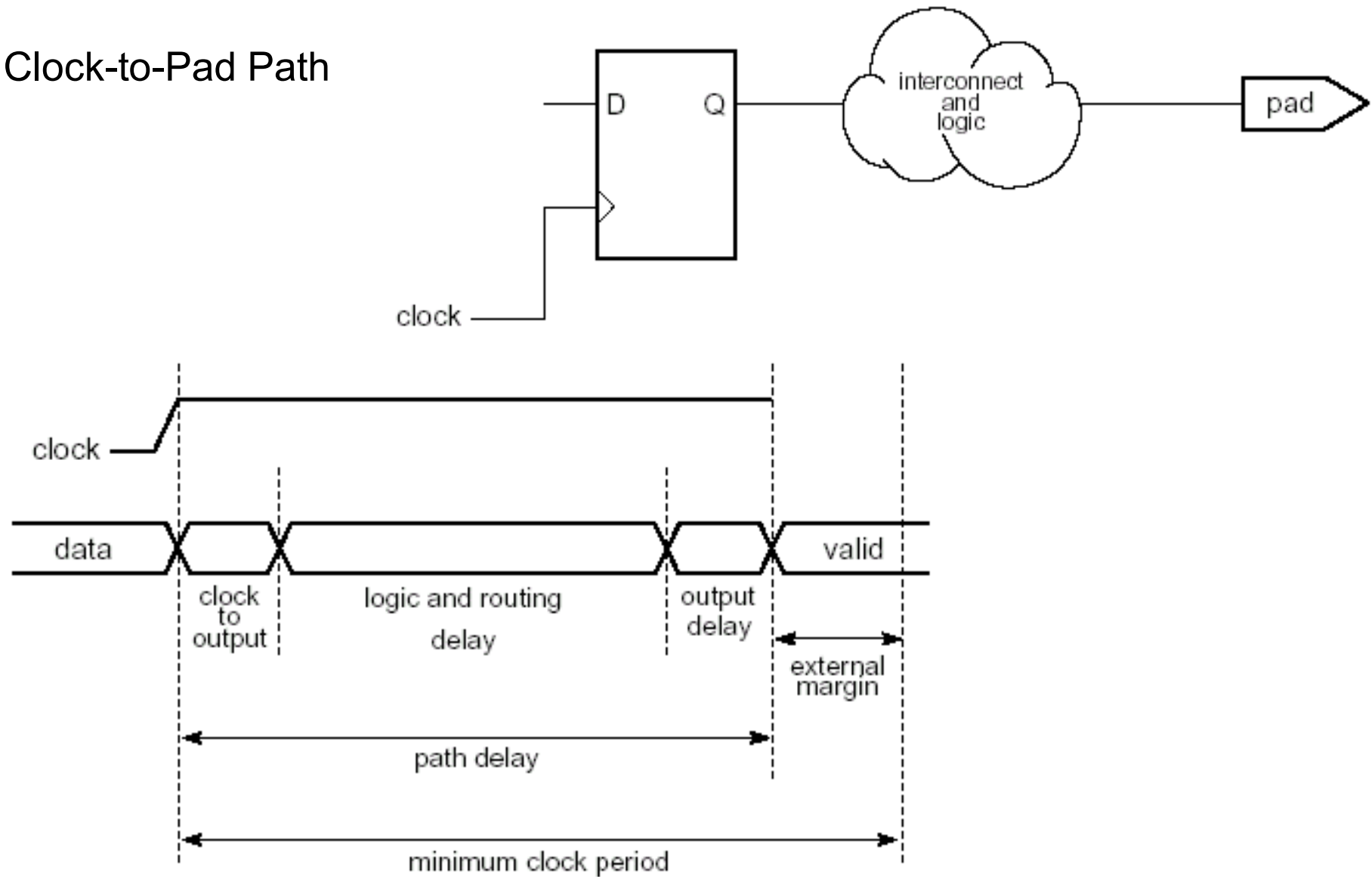
# 时钟到建立(Clock-to-Setup Path)

Falling to Rising Edge



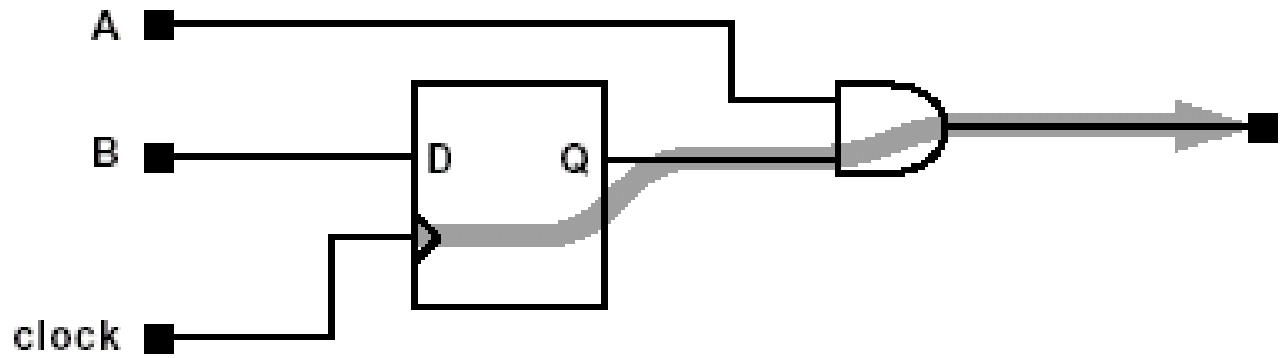
# 时钟到管脚(Clock-to-Pad Path)

Clock-to-Pad Path

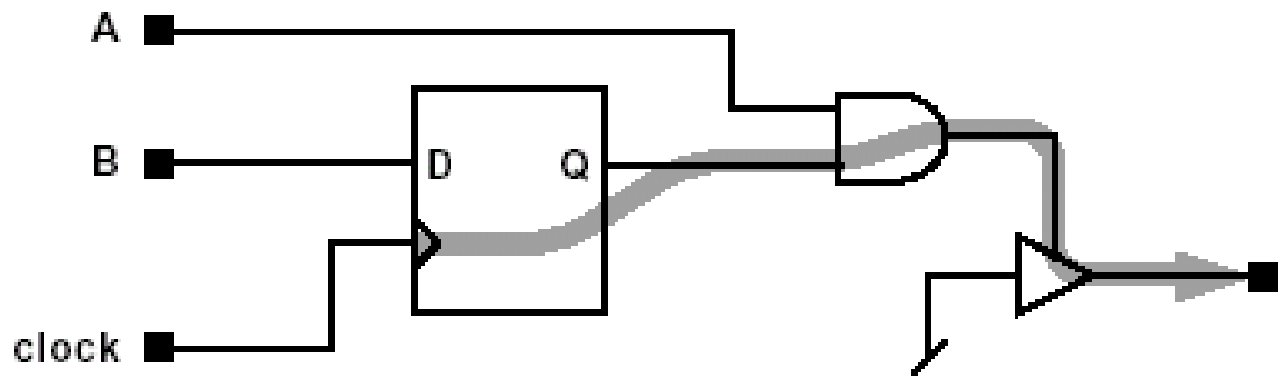


# 时钟到管脚(Clock-to-Pad Path)

Clock-to-Pad Path



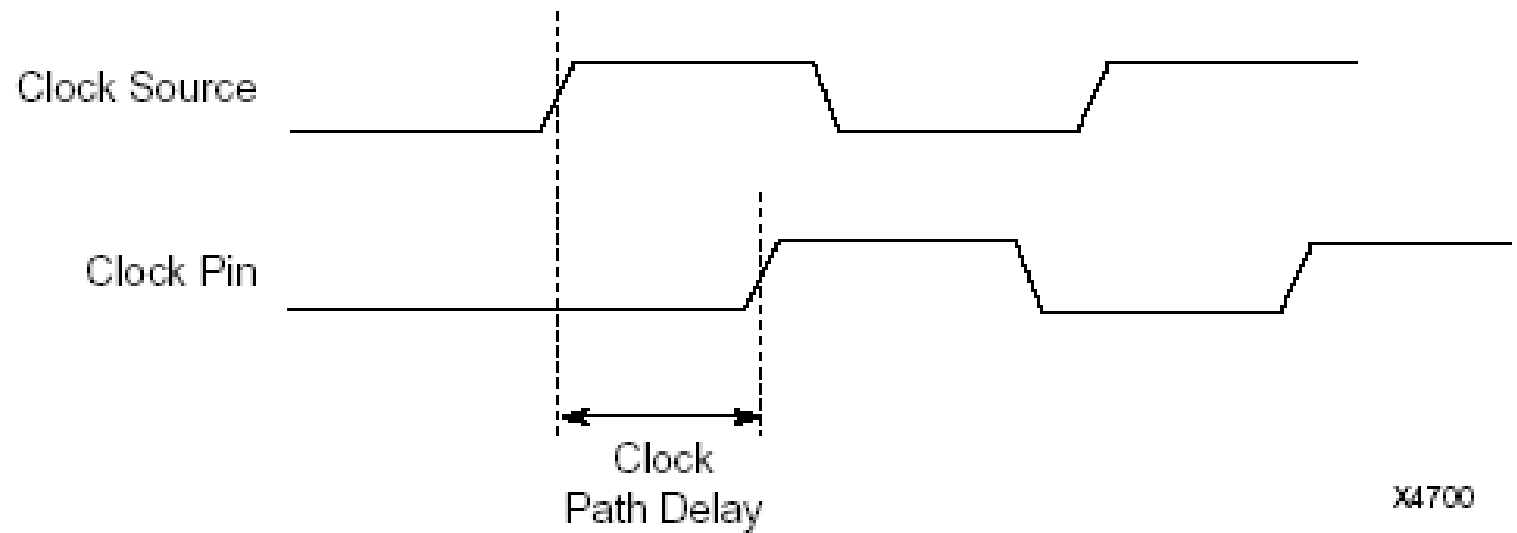
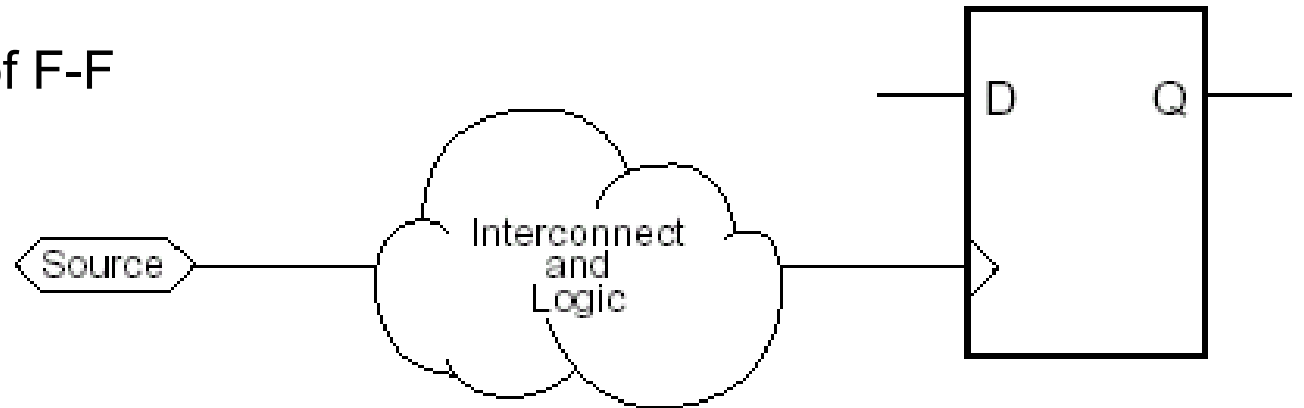
通过三态控制管脚的情况





# 结束于时钟引脚(Paths Ending at Clock Pin of Filp-Flops)

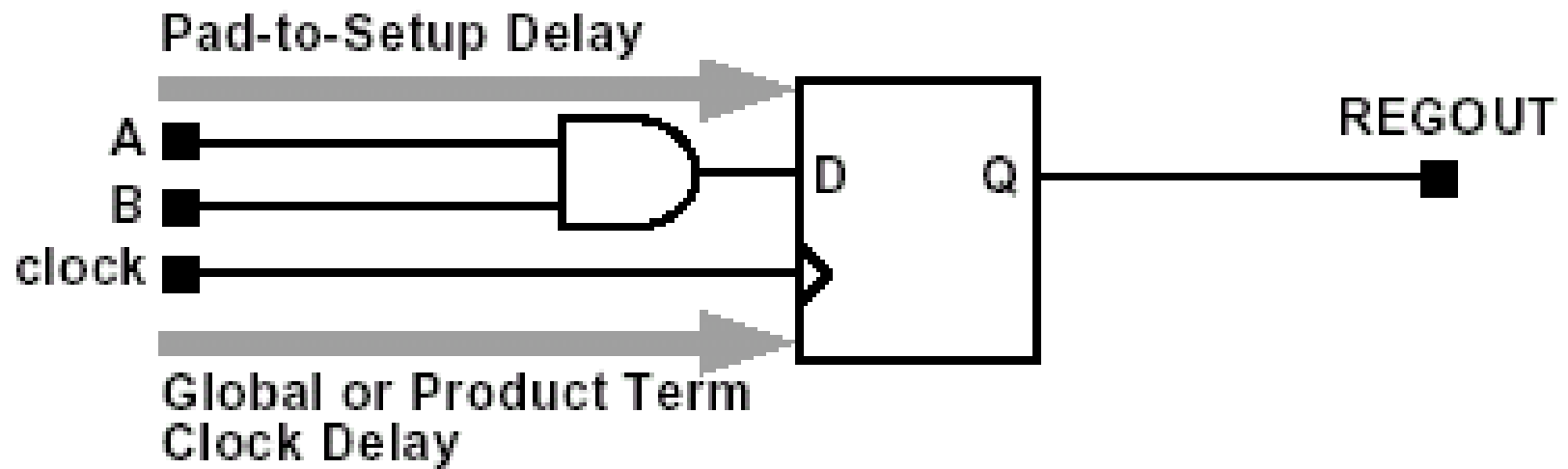
Ending at Clock Pin of F-F



# 管脚上的建立到时钟(Setup to Clock at the Pad)

---

Setup to Clock at the Pad



# 时钟管脚到输出管脚(Clock Pad to Output Pad)

---

Clock Pad to Output Pad

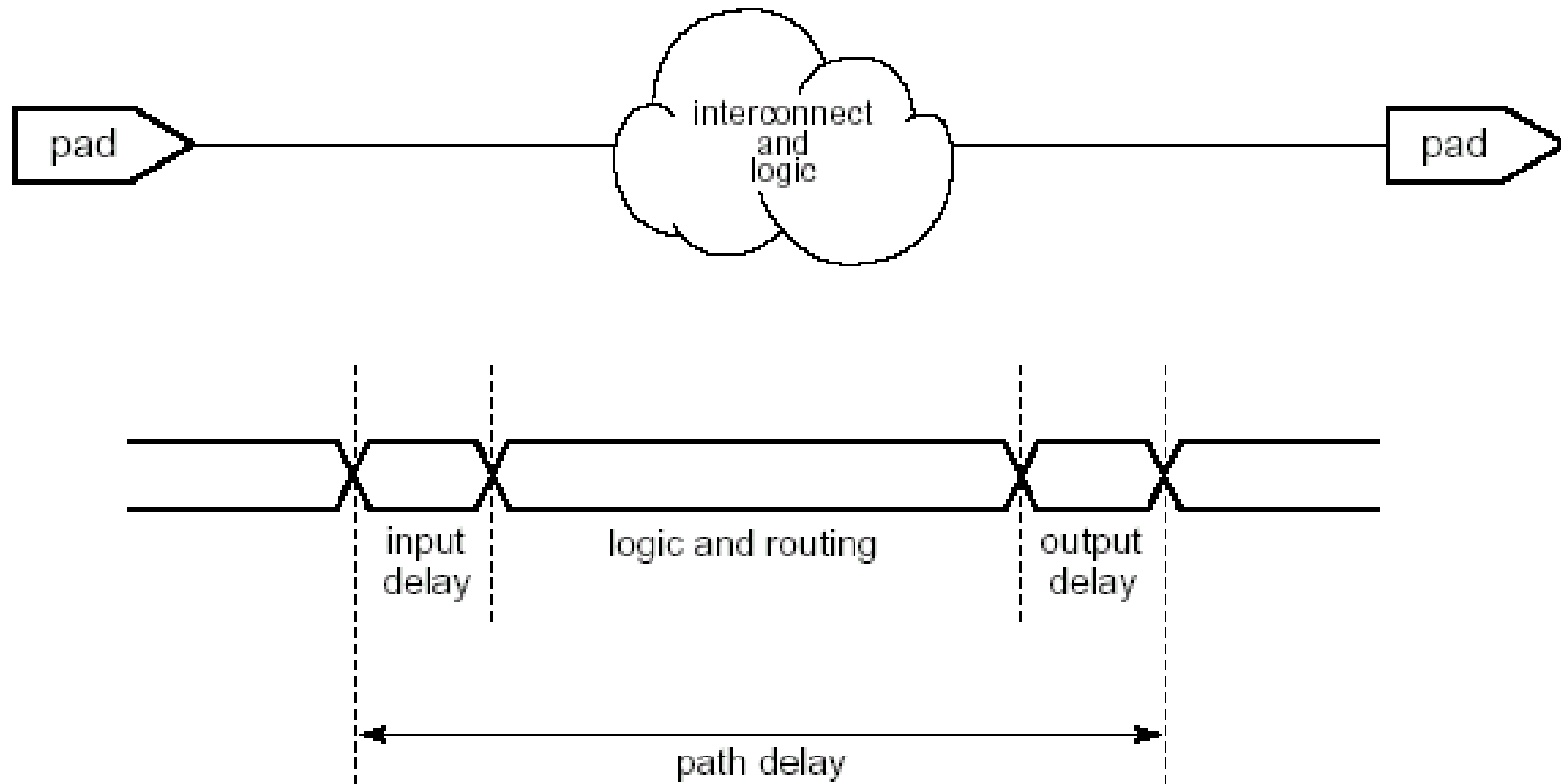
时钟管脚到输出管脚

# 管脚到管脚(Pad to Pad)

---

Pad to Pad

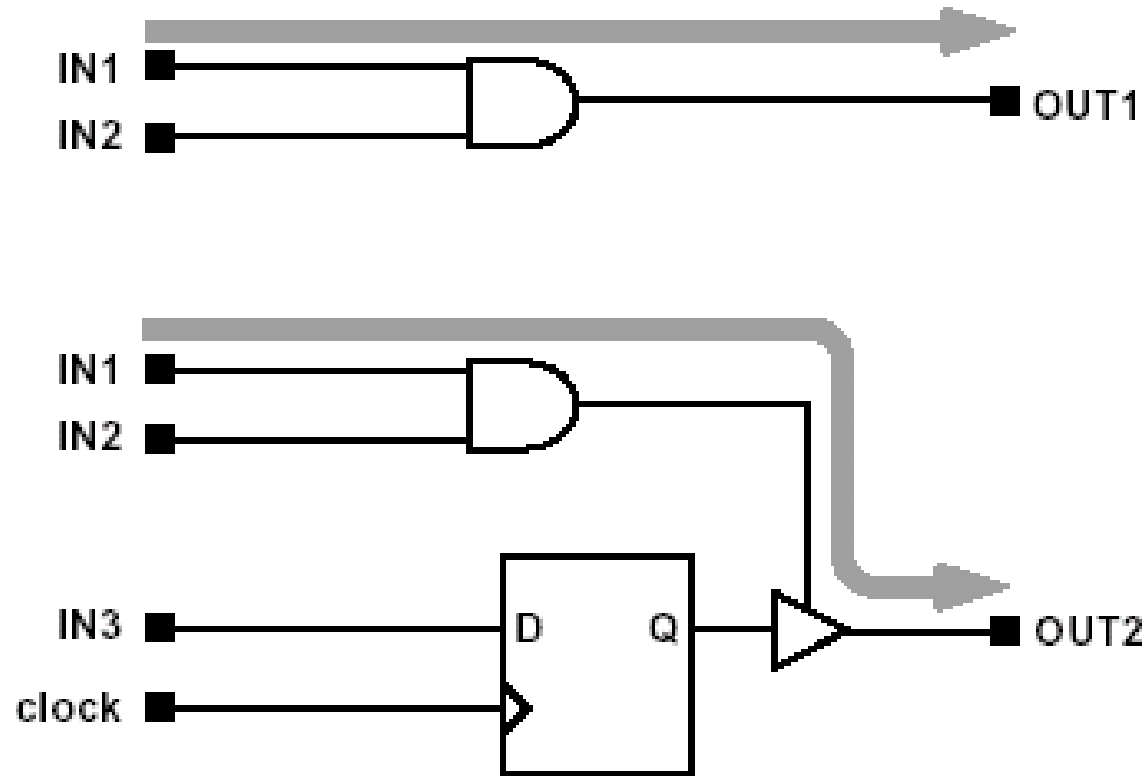
t<sub>PD</sub> – 管脚到管脚时延



# 管脚到管脚(Pad to Pad)

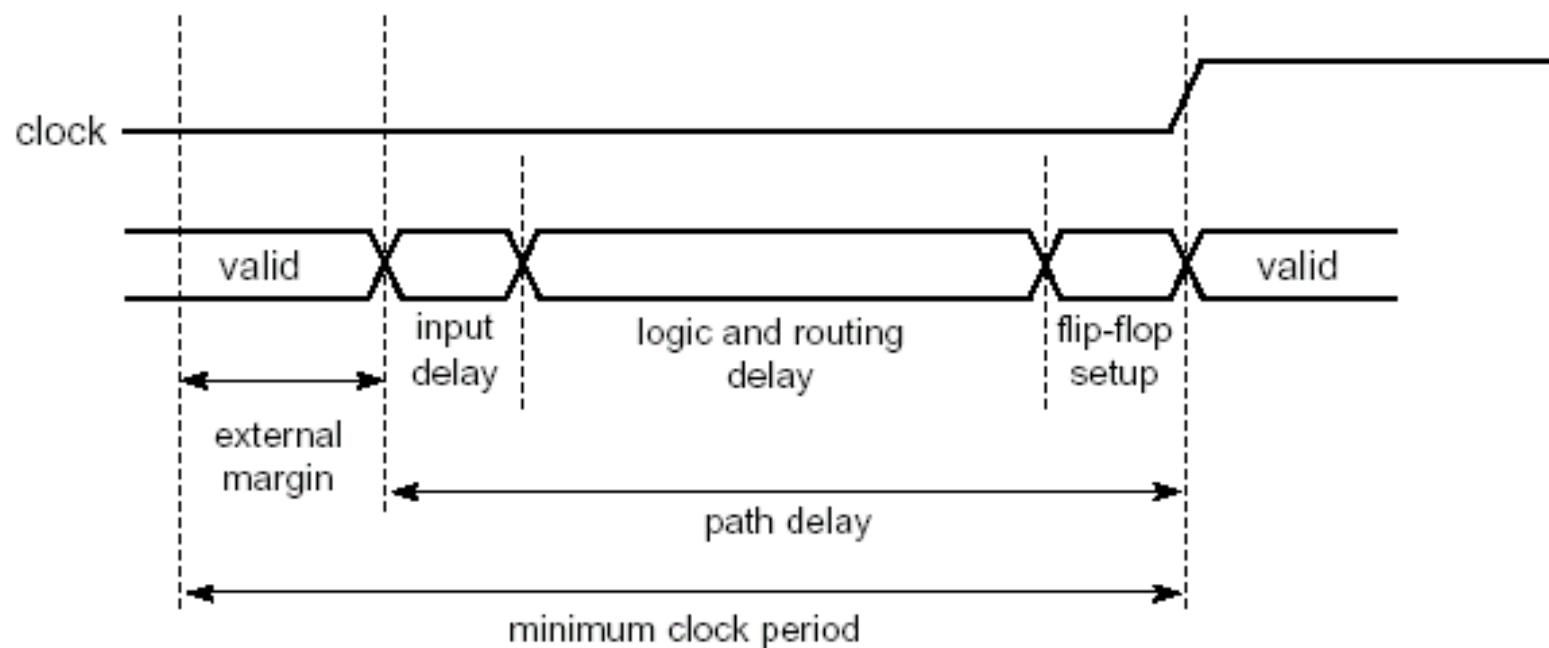
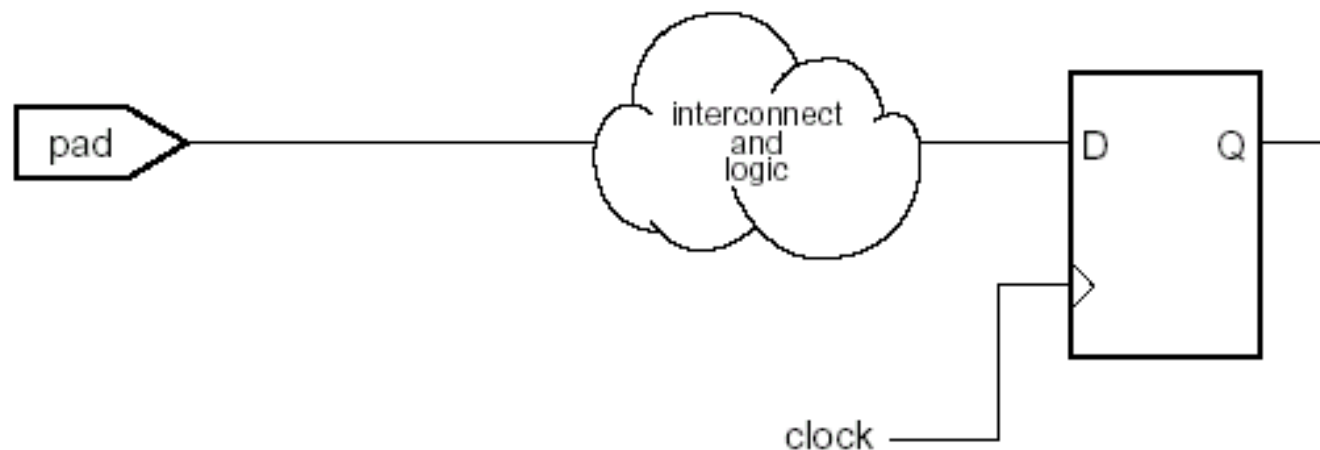
---

Pad to Pad Path



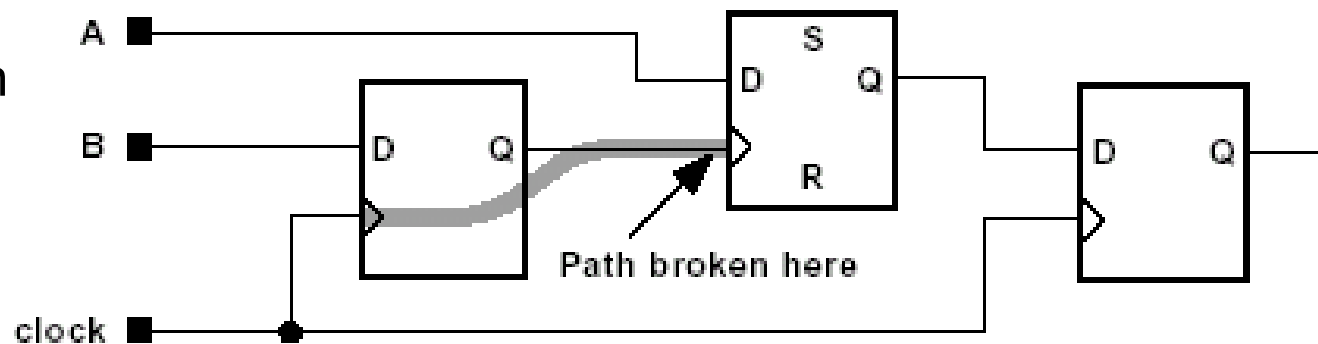
# 管脚到建立(Pad to Setup)

Pad to Setup

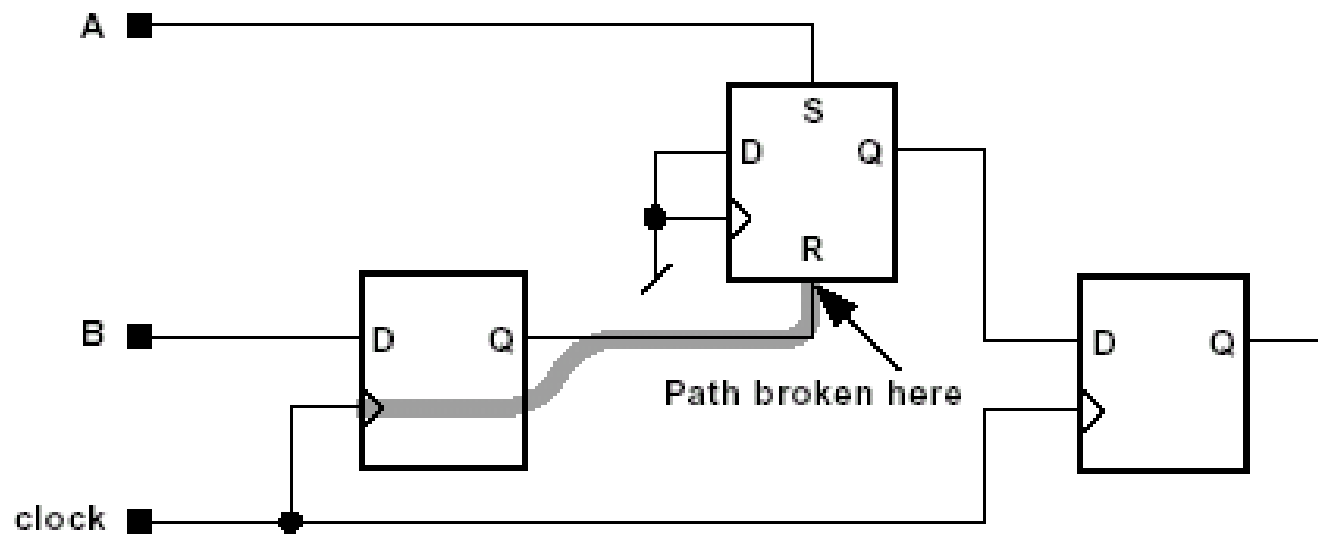


# 问题讨论—Clock-to-Setup Path

Clock-to-Setup Path

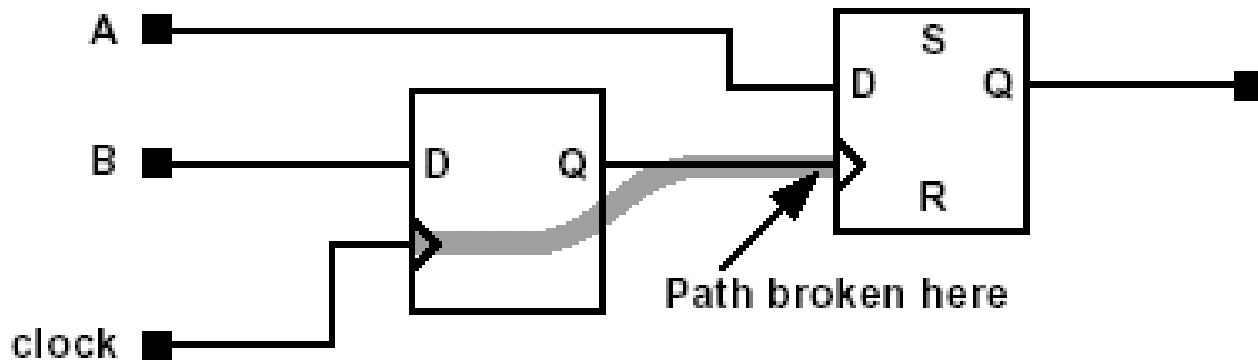


时序分析路径打断

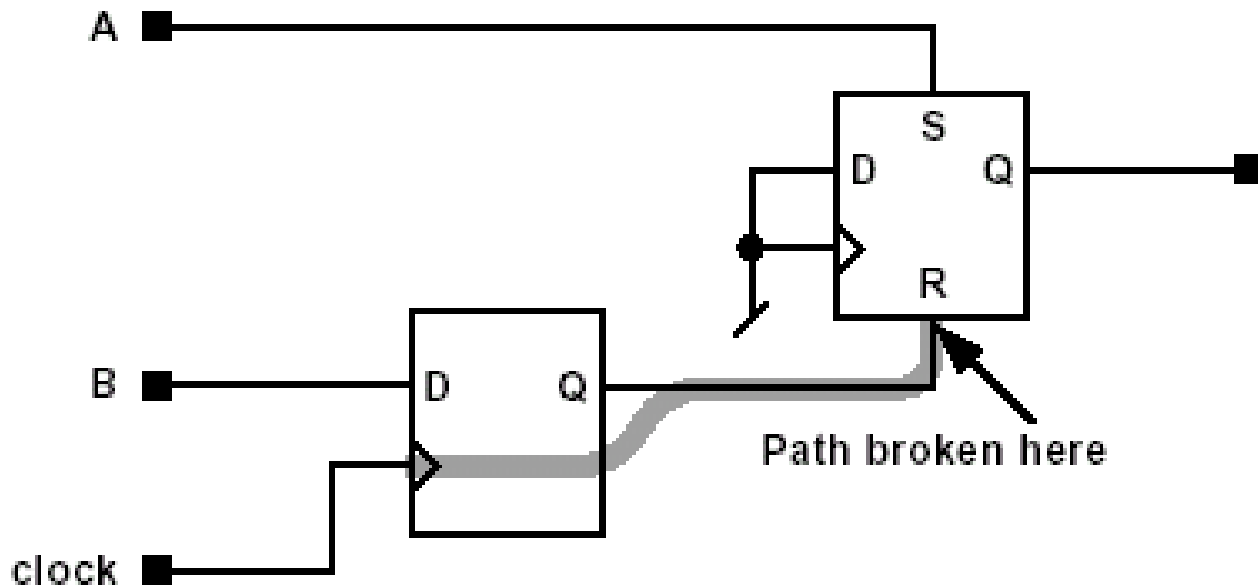


# 问题讨论—Clock-to-Pad Path

Clock-to-Pad Path



时序分析路径打断

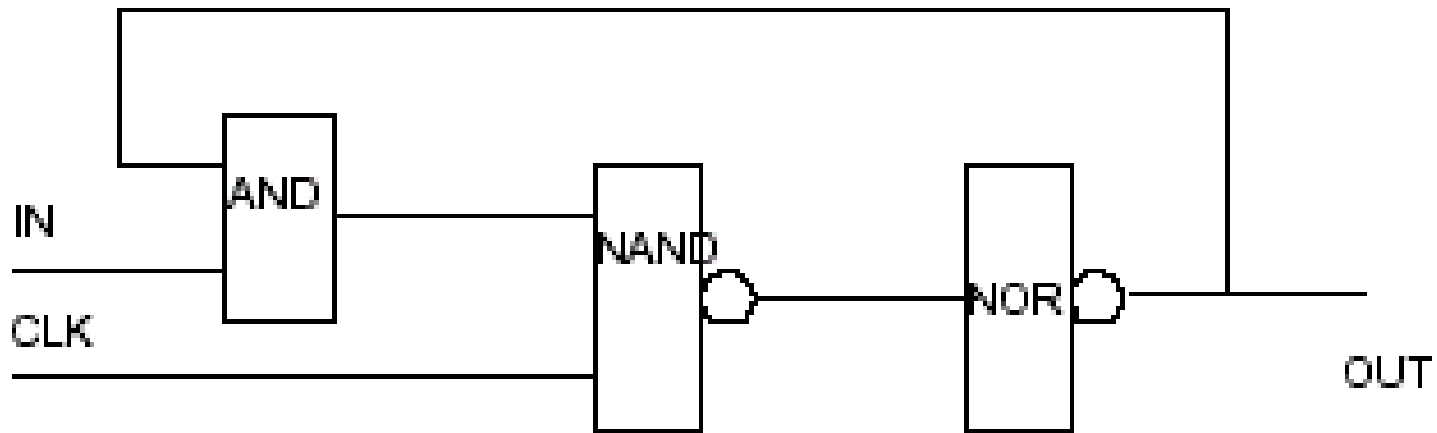




# 问题讨论—反馈环(Feedback Loops)

---

## Feedback Loops



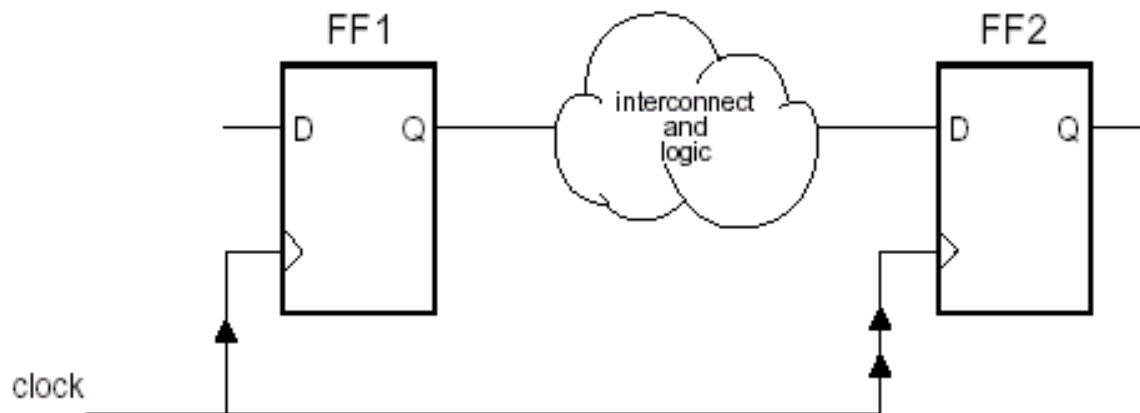
时序分析路径打断

尽量使用同步逻辑来替代完  
成功能

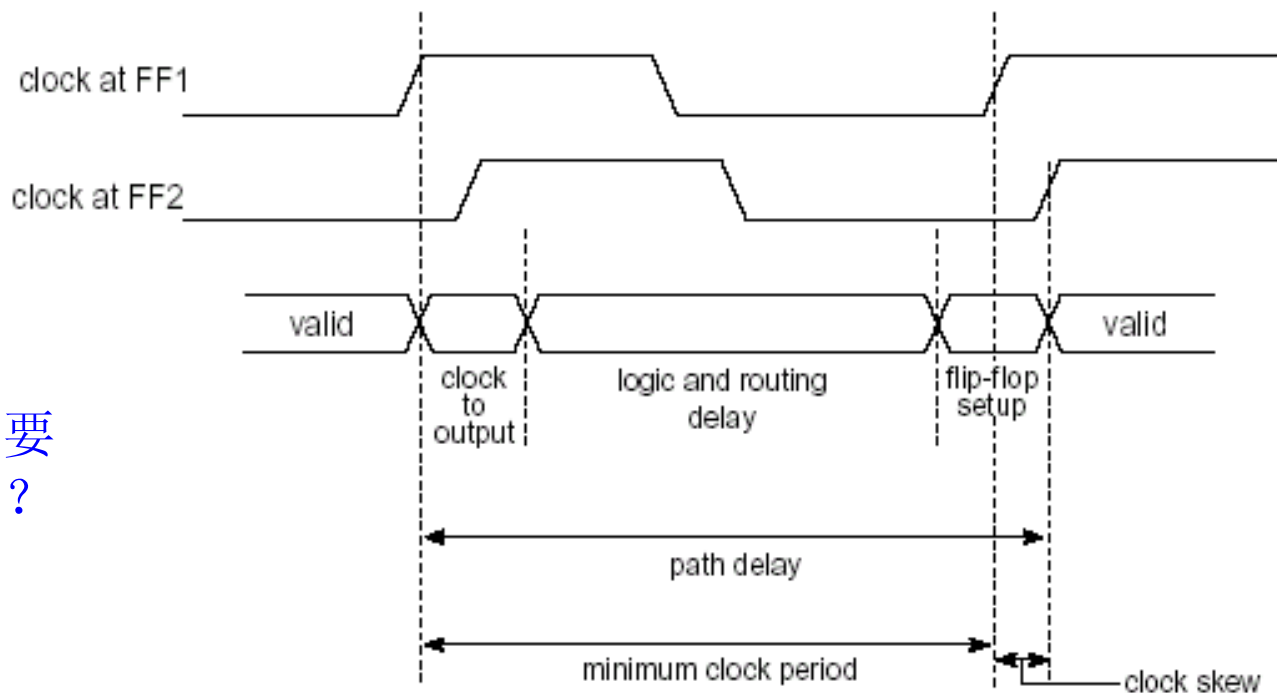
# 问题讨论—时钟偏移(Clock Skew)

Clock Skew

时钟偏移



为什么规范要求用全局钟?



# 片外时延(**Off-Chip Delay**)

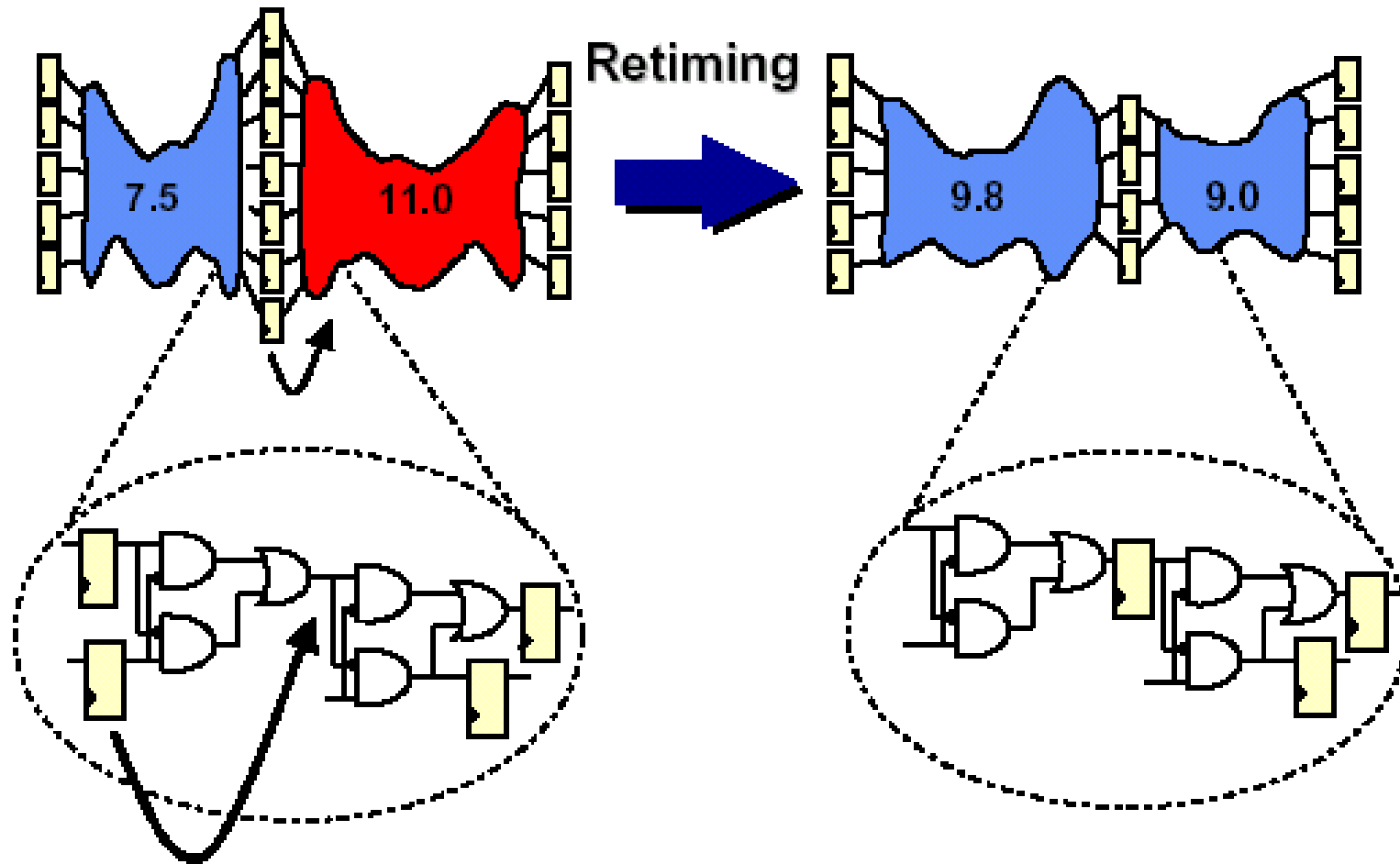
---

## Off-Chip Delay

为了决定系统时钟速度，必须指定系统时钟的外部时延。这样，在静态时序分析的时候，可以将此时延计算在内。

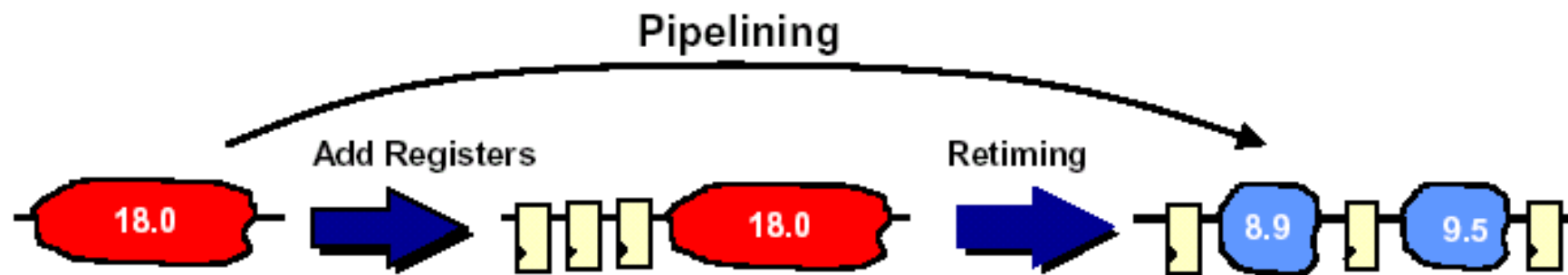
# 时序优化方法—Retiming

Clock Period = 10 ns



# 时序优化方法—Pipelining

---

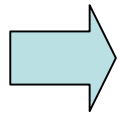


# 课程目录

---

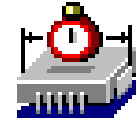
◆ 静态时序分析—概念与流程

◆ 静态时序分析—时序路径



◆ 静态时序分析—分析工具

# 静态时序分析工具—Timing Analyzer



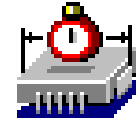
## Timing Analyzer

```
✓ Delay Matrix
  Setup/Hold Matrix
  Registered Performance
```

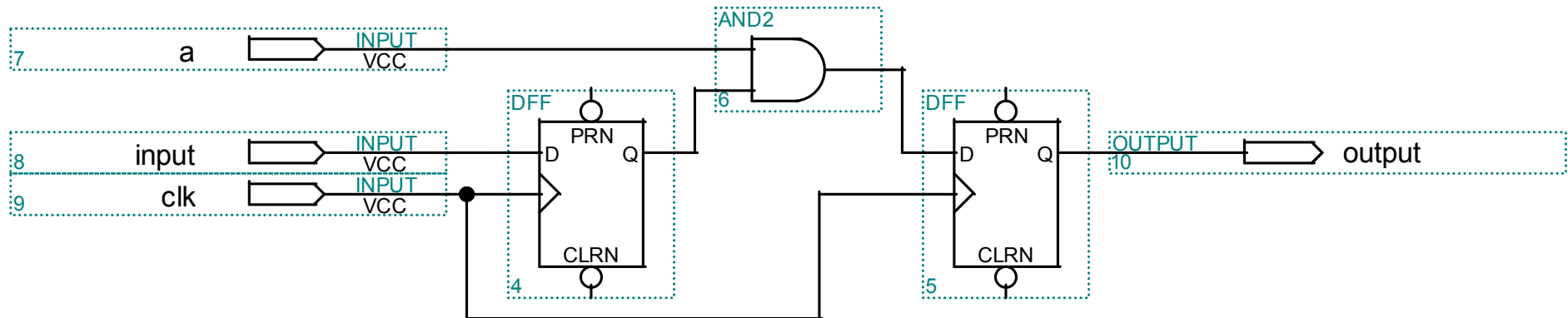
Altera公司MAX+plus II自带的静态时序分析工具，可以进行：

- ◆ 时序路径的时延分析（Delay Matrix）
- ◆ 建立/保持时间分析（Setup/Hold）\*\*
- ◆ 同步逻辑性能（Registered Performance）

# 静态时序分析工具—Timing Analyzer



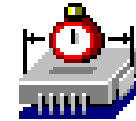
## Timing Analyzer – 分析实例



器件型号: EPM7256AEQC208-10



# 静态时序分析工具—Timing Analyzer



## Timing Analyzer – Delay Matrix

### 路径时延分析

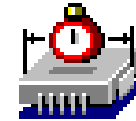
可分析多个源和多个目标结点（node）之间的路径传播时延，

		Destination		
		output	:4.D	:5.D
Source	a			7.2ns
	clk	6.4ns		10.8ns
	input		7.2ns	
	:4.CLK			7.6ns
	:4.Q			6.0ns
	:5.CLK	3.2ns		
	:5.Q	1.6ns		

0 50 100

Start Stop List Paths

# 静态时序分析工具—Timing Analyzer



## Timing Analyzer – Setup/Hold

### 建立/保持时间分析

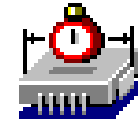
计算输入引脚到DFF的数据、时钟、时钟使能输入端的最小要求建立、保持时间

Clocks			
	clk		
Inputs	:4.Q input	6.9ns/0.0ns	
	:5.Q a	6.9ns/0.0ns	

0 50 100

Start Stop List Paths

# 静态时序分析工具—Timing Analyzer

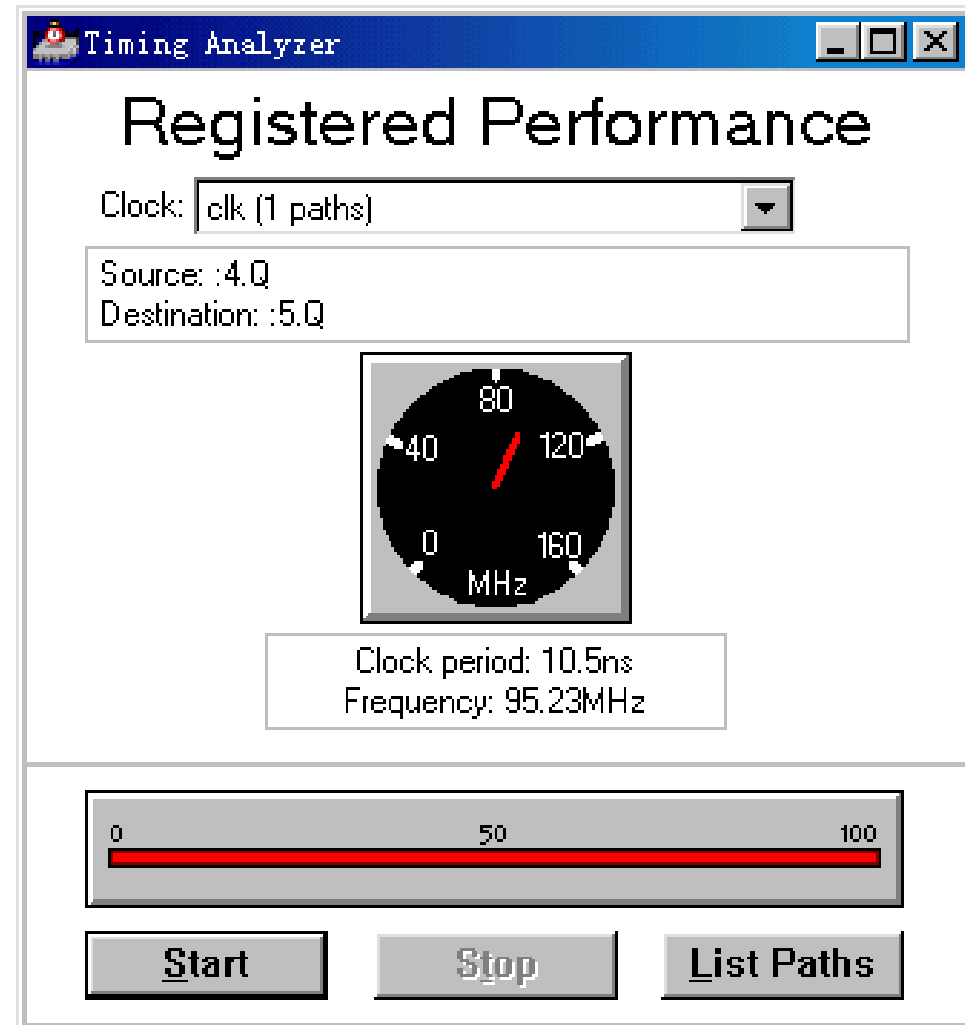


## Timing Analyzer – Registered Performance

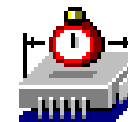
### 同步逻辑性能分析

分析同步逻辑，确定限制性能的时延、最小时钟周期、最大时钟频率。

实际上包含了各内部寄存器的建立、保持时间分析。



# 静态时序分析工具—Timing Analyzer



## Timing Analyzer – Registered Performance

### 10.5s 是怎样算出来的？

根据同步逻辑的时延模型， $T = t_{CO} + t_{DELAY} + t_{SU}$

查MAX7000A/E datasheet, EPM7256AEQC208-10器件,

$t_{CO}$ 在手册内相当于参数 $t_{RD}$ , 最小值为1.6ns;

$t_{SU}$ 在手册内相当于参数 $t_{SU}$ , 最小值为2.9ns;

在此处, 很显然寄存器4:Q和寄存器5:D之间的路径为关键路径, 则 $t_{DELAY}$ 相当于4:Q和5:D之间的时延, 查Delay Matrix分析结果,  $t_{DELAY}$ 为6.0ns。

$$T = t_{CO} + t_{DELAY} + t_{SU} = 1.6\text{ns} + 6.0\text{ns} + 2.9\text{ns} = 10.5\text{ns}$$

## 静态时序分析工具—**PrimeTime**

目前不要求掌握，只要了解基本思路

主要用于大规模逻辑设计

## 静态时序分析工具—PrimeTime

---

PrimeTime是Synopsys的静态时序分析工具，为业界标准，占居最大的市场份额。

PrimeTime是数字ASIC设计的sign-off必选工具，受到所有EDA工具和IC家的支持。

FPGA逻辑静态时序分析，仅用到PrimeTime的一小部分功能。

# 静态时序分析报告术语

---

## 信号到达时间 (**Arrival Time**)

表示实际计算所得的信号到达逻辑电路中某一点的绝对时间等于信号到达某条路径起点的时间加上信号在该条路径上的逻辑单元间传递延时的总和

## 要求到达时间 (**Required Arrival Time**)

简称RAT 表示要求信号在逻辑电路的某一特定点处的到达时间

## 迟缓 (**Slack**)

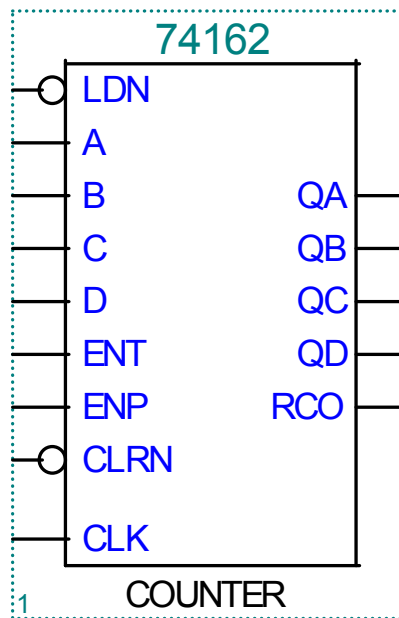
表示在逻辑电路的某一特定点处要求到达时间与实际到达时间之间的差。**Slack**的值表示该信号到达得是否太晚或太早

# 课程练习1

---

在MAX+plus II中，使用Timing Analyzer分析一下74162的最高工作频率。（可打开74162的内部电路，然后进行分析）

目标器件为：EPF6024AQC240-3





## 课程练习2

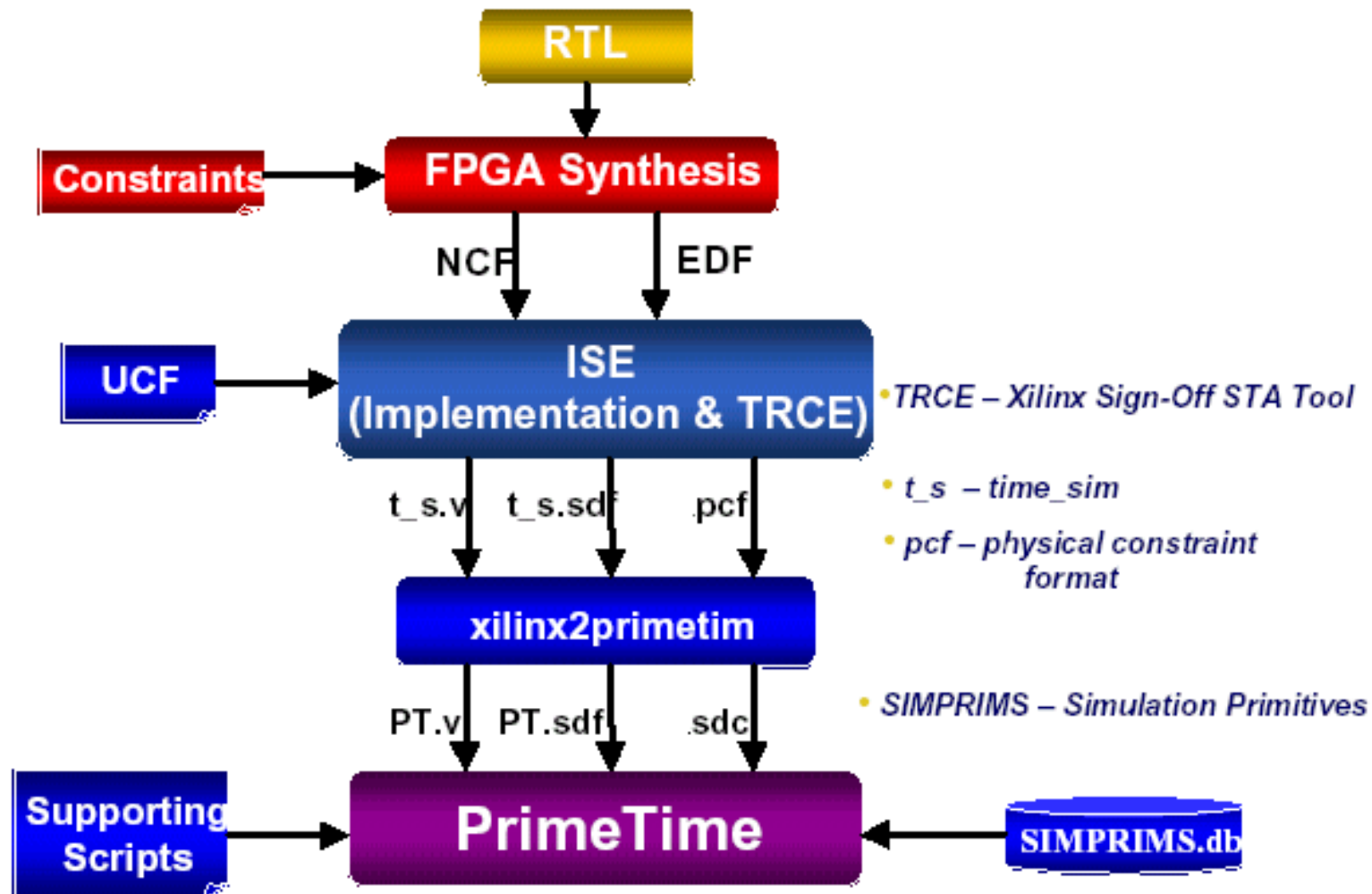
---

在MAX+plus II中，使用Timing Analyzer分析**TMU**单板逻辑内部时钟的最大工作频率是否满足要求。

目标器件为：EPF6024AQC240-3

# 静态时序分析工具—PrimeTime

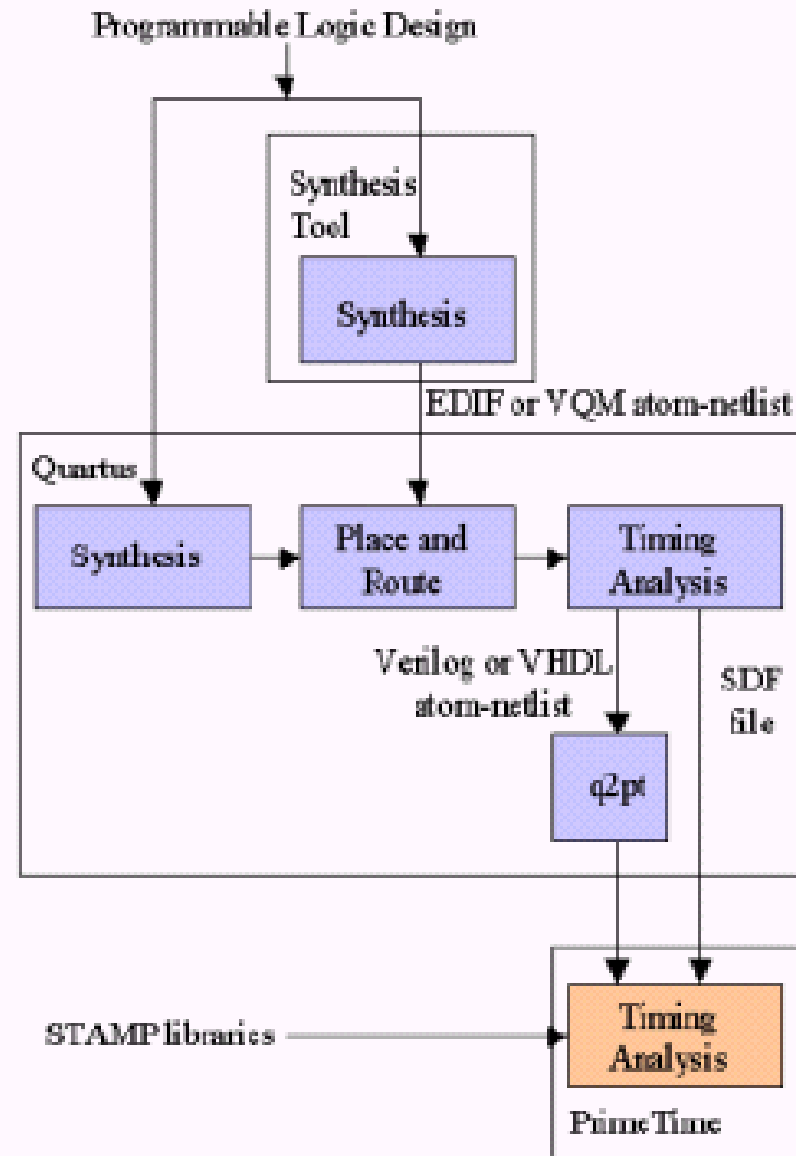
## PrimeTime与Xilinx设计流程



# 静态时序分析工具—PrimeTime

## PrimeTime与Altera设计流程

为了支持大规模的逻辑验证，可编程逻辑器件厂商（如Altera、Xilinx）的工具支持PrimeTime。



# 静态时序分析工具—PrimeTime

---

## PrimeTime与Altera设计流程—Script 实例

```
set search_path ". $QUARTUS_ROOTDIR/eda/synopsys/primetime/lib"  
  
set link_path {* alt_vtl.db apex20ke_asynch_mem_lib.db  
apex20ke_lvds_receiver_lib.db  
  
apex20ke_cam_lib.db apex20ke_lvds_transmitter_lib.db apex20ke_io_lib.db  
apex20ke_pll_lib.db  
  
apex20ke_lcell_lib.db apex20ke_pterm_lib.db}  
  
read_verilog  
{ $QUARTUS_ROOTDIR/eda/synopsys/primetime/lib/apex20ke_camslice_pt.v }  
  
read_verilog  
{ $QUARTUS_ROOTDIR/eda/synopsys/primetime/lib/apex20ke_ramslice_pt.v }  
  
read_verilog snug_pt.vo  
  
current_design snug  
  
link_design snug  
  
read_sdf snug_v.sdo
```

# 静态时序分析工具—PrimeTime

---

## PrimeTime与Altera设计流程—Script 实例

```
create_clock "CLK" -period 4 -waveform {0 2}
```

```
check_timing
```

```
report_analysis_coverage
```

```
report_timing
```

# 静态时序分析工具—PrimeTime

---

## PrimeTime时序约与Xilinx约束对比

### FPGA时序分析常用项

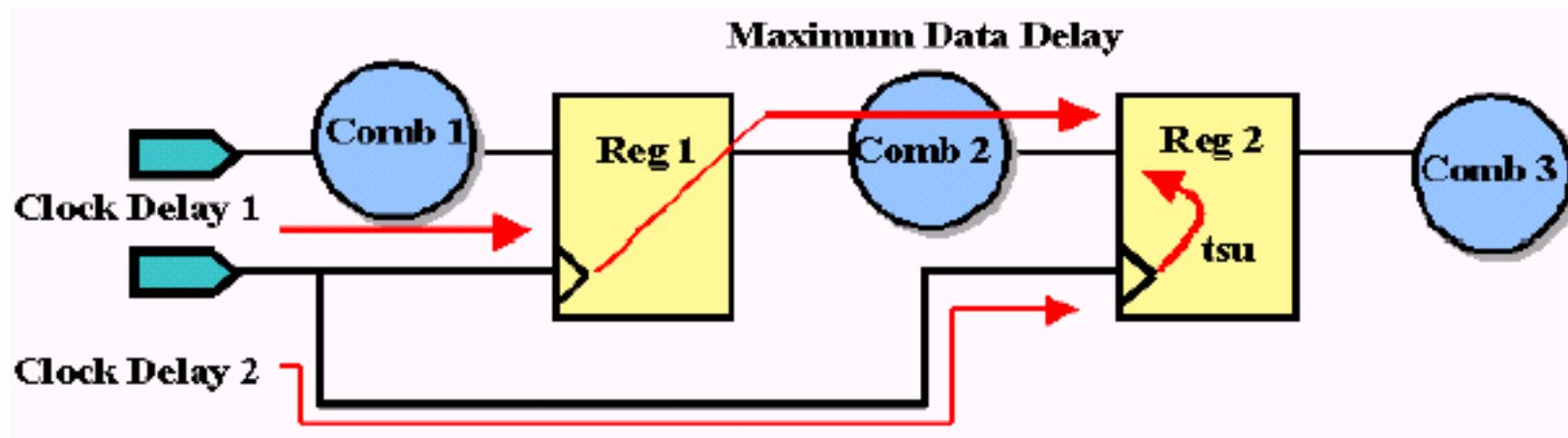
	SDC时序约束	UCF时序约束
时钟周期	create_clock	PERIOD
输出时延	set_input_delay	OFFSET IN
输出时延	set_output_delay	OFFSET OUT
最大时延	set_max_delay	MAXDELAY
伪路径	set_false_path	TIG

SDC – Synopsys Design Constraints

UCF– Xilinx User Constraints File

# 静态时序分析工具—PrimeTime

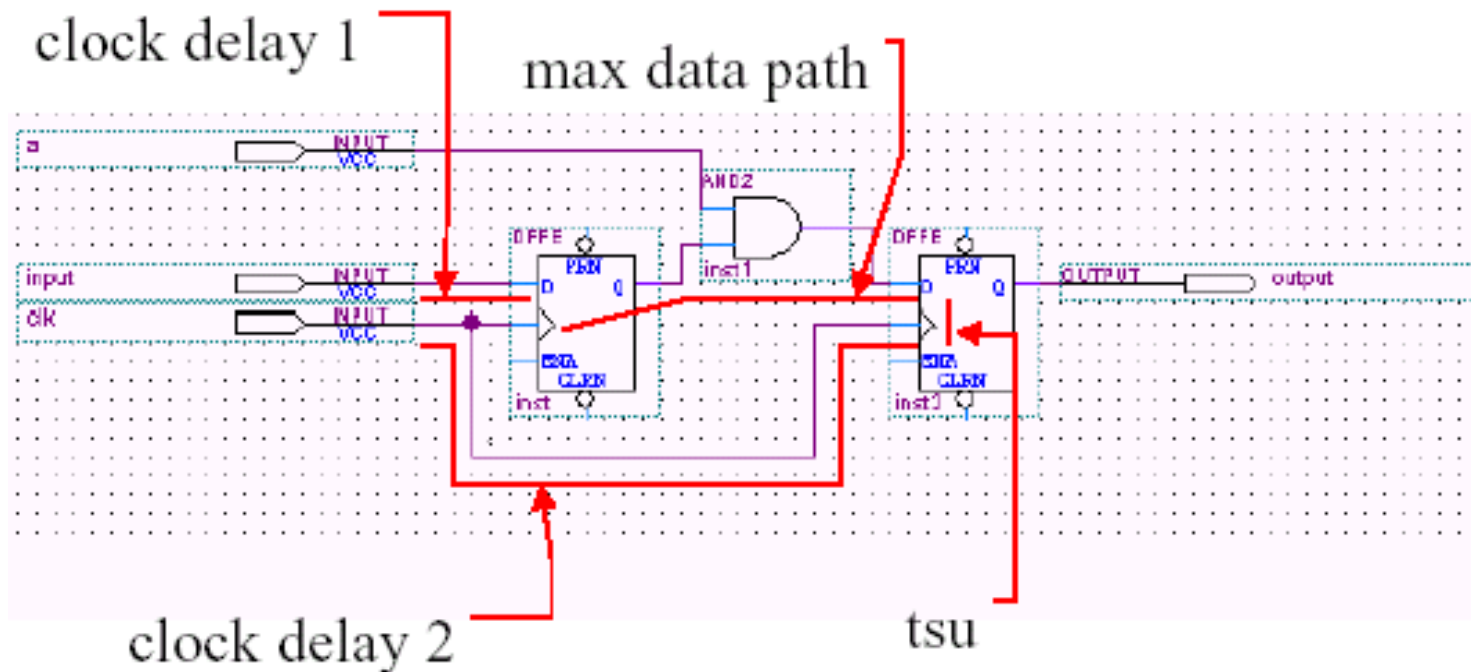
## 建立时间检查



$$\text{clock delay 1} + \text{max data path} - \text{clock delay 2} + \text{tsu} \leq \text{clock period}$$

# 静态时序分析工具—PrimeTime

## 建立时间检查—实例



- 对全局时钟（Global Clock）来说，Clock delay 可认为是0ns
- Max data path是由寄存器的 tCO时间加上两个寄存器之间的路径时延
- tsu是触发器的建立时间



# 静态时序分析工具—PrimeTime

---

## 建立时间检查—实例

- **Clock Delay 1 = 0ns**
- **Clock Delay 2 = 0ns**
- **Max Data Path = tco of source reg + path between reg = 1.449ns + .258ns = 1.707ns**
- **With 4ns clock period, the slack is 4ns-1.707ns = 2.293ns**

# 静态时序分析工具—PrimeTime

## 建立时间检查—实例

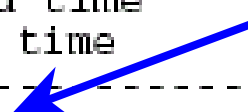
## PrimeTime分析报告

Startpoint: i\_inst (rising edge-triggered flip-flop clocked by clk)  
Endpoint: i\_inst2 (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path Type: max

Point	Incr	Path
-----		
clock clk (rise edge)	0.000	0.000
clock network delay (ideal)	0.000	0.000
i_inst/clk (apex20ke_lcell)	0.000	0.000 r
i_inst/regout (apex20ke_lcell)	1.449 *	1.449 r
i_inst2/datad (apex20ke_lcell)	0.258 *	1.707 r
data arrival time		1.707
clock clk (rise edge)	4.000	4.000
clock network delay (ideal)	0.000	4.000
i_inst2/clk (apex20ke_lcell)		4.000 r
library setup time	0.000	4.000
data required time		4.000
-----		
data required time		4.000
data arrival time		-1.707
-----		
slack (MET)		2.293

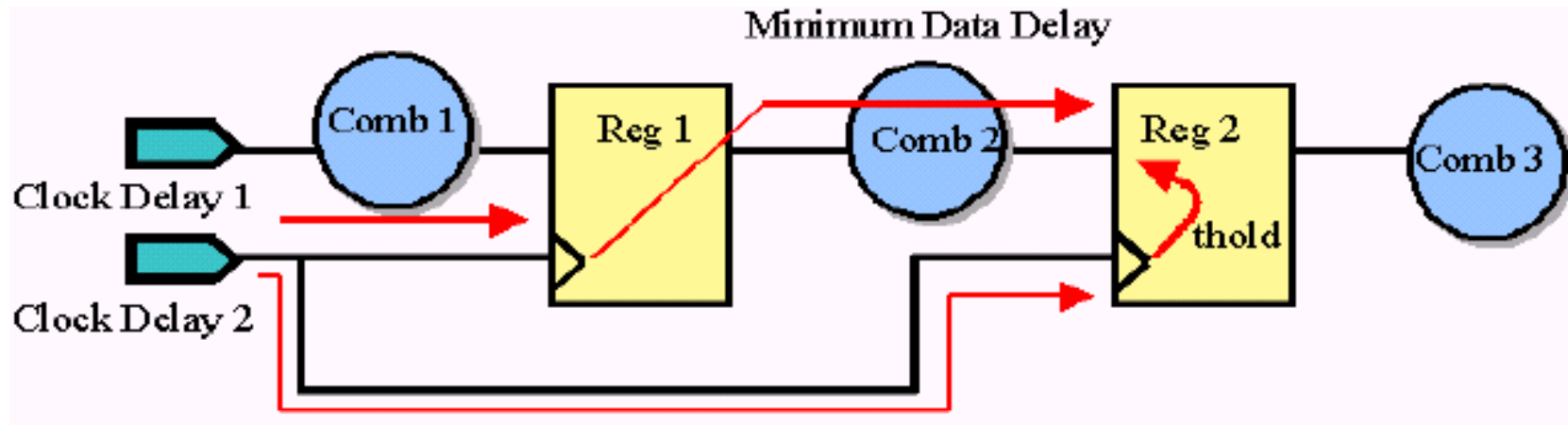
MET: 满足建立时间

VIOLATED: 不满足



# 静态时序分析工具—PrimeTime

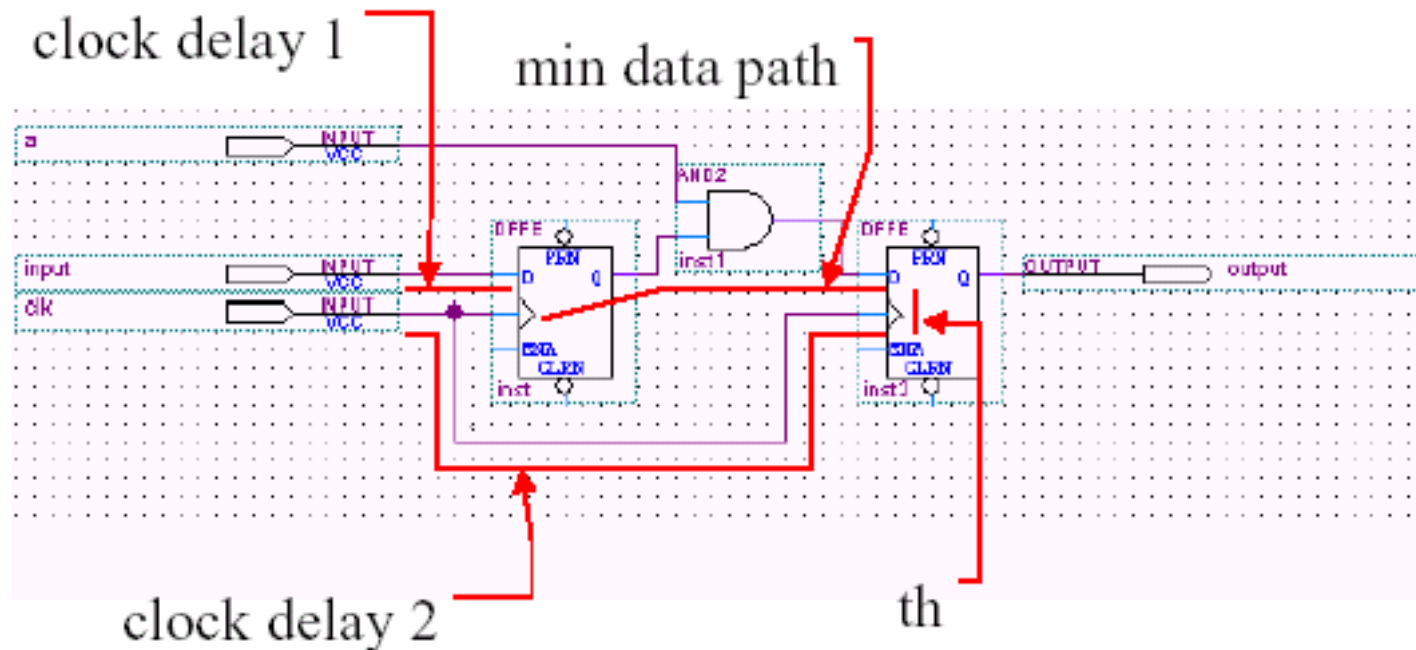
## 保持时间检查



$$\text{clock delay 1} + \text{minimum data delay} - \text{clock delay 2} - \text{th} \geq 0$$

# 静态时序分析工具—PrimeTime

## 保持时间检查—实例



- 对全局时钟（Global Clock）来说，Clock delay 可认为是0ns
- Min data path是由寄存器的 tCO时间加上两个寄存器之间的路径时延
- th是触发器的建立时间

## 静态时序分析工具—PrimeTime

---

### 保持时间检查—实例

- **Clock Delay 1 = 0ns**
- **Clock Delay 2 = 0ns**
- **Minimum Data Path = tco of source reg + path  
between reg = 1.449ns + .258ns = 1.707ns**
- **Intrinsic hold time = 1.284ns**
- **Slack is 1.707ns-1.284ns = 0.493ns**

# 静态时序分析工具—PrimeTime

## 保持时间检查—实例

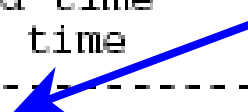
## PrimeTime分析报告

Startpoint: i\_inst (rising edge-triggered flip-flop clocked by clk)  
Endpoint: i\_inst2 (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path Type: min

Point	Incr	Path
-----		
clock clk (rise edge)	0.000	0.000
clock network delay (ideal)	0.000	0.000
i_inst/clk (apex20ke_lcell)	0.000	0.000 r
i_inst/regout (apex20ke_lcell)	1.449 *	1.449 r
i_inst2/datad (apex20ke_lcell)	0.258 *	1.707 r
data arrival time		1.707
clock clk (rise edge)	0.000	0.000
clock network delay (ideal)	0.000	0.000
i_inst2/clk (apex20ke_lcell)		0.000 r
library hold time	1.284 *	1.284
data required time		1.284
-----		
data required time		1.284
data arrival time		-1.707
-----		
slack (MET)		0.423

MET: 满足建立时间

VIOLATED: 不满足



谢谢！