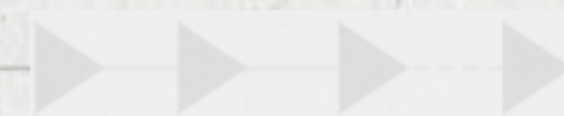


芯动力——硬件加速设计方法

第四章 逻辑综合(3)

邸志雄@西南交通大学

zxdi@home.swjtu.edu.cn



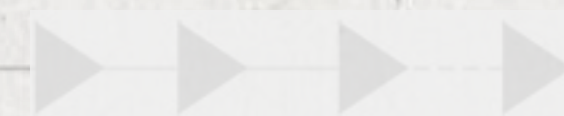
施加约束

- Design Compiler 是一个约束驱动(constrain-driven)的综合工具，它的结果是与设计者施加的约束条件密切相关的。

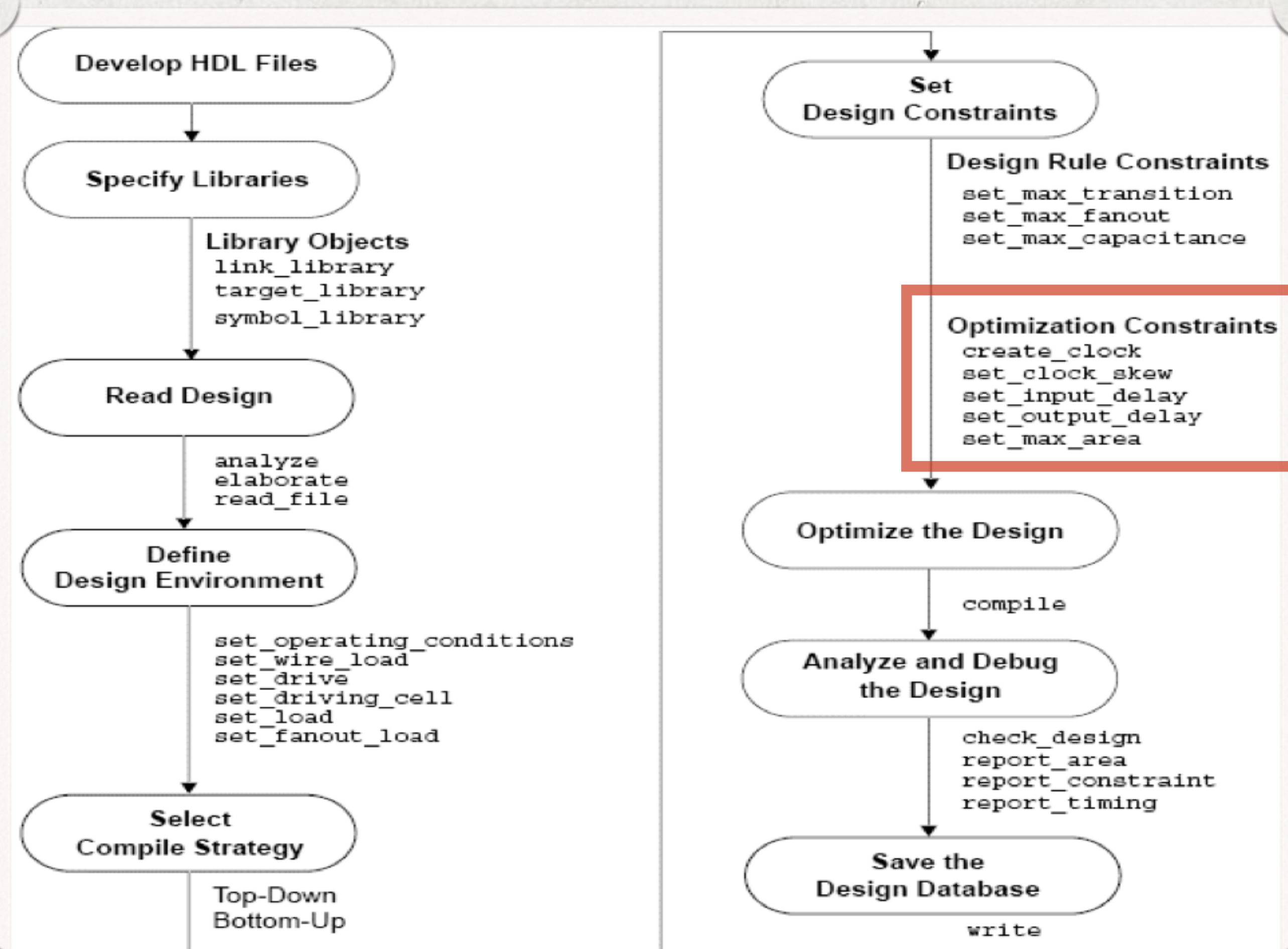
我们主要讨论怎样给电路施加约束条件

时序约束

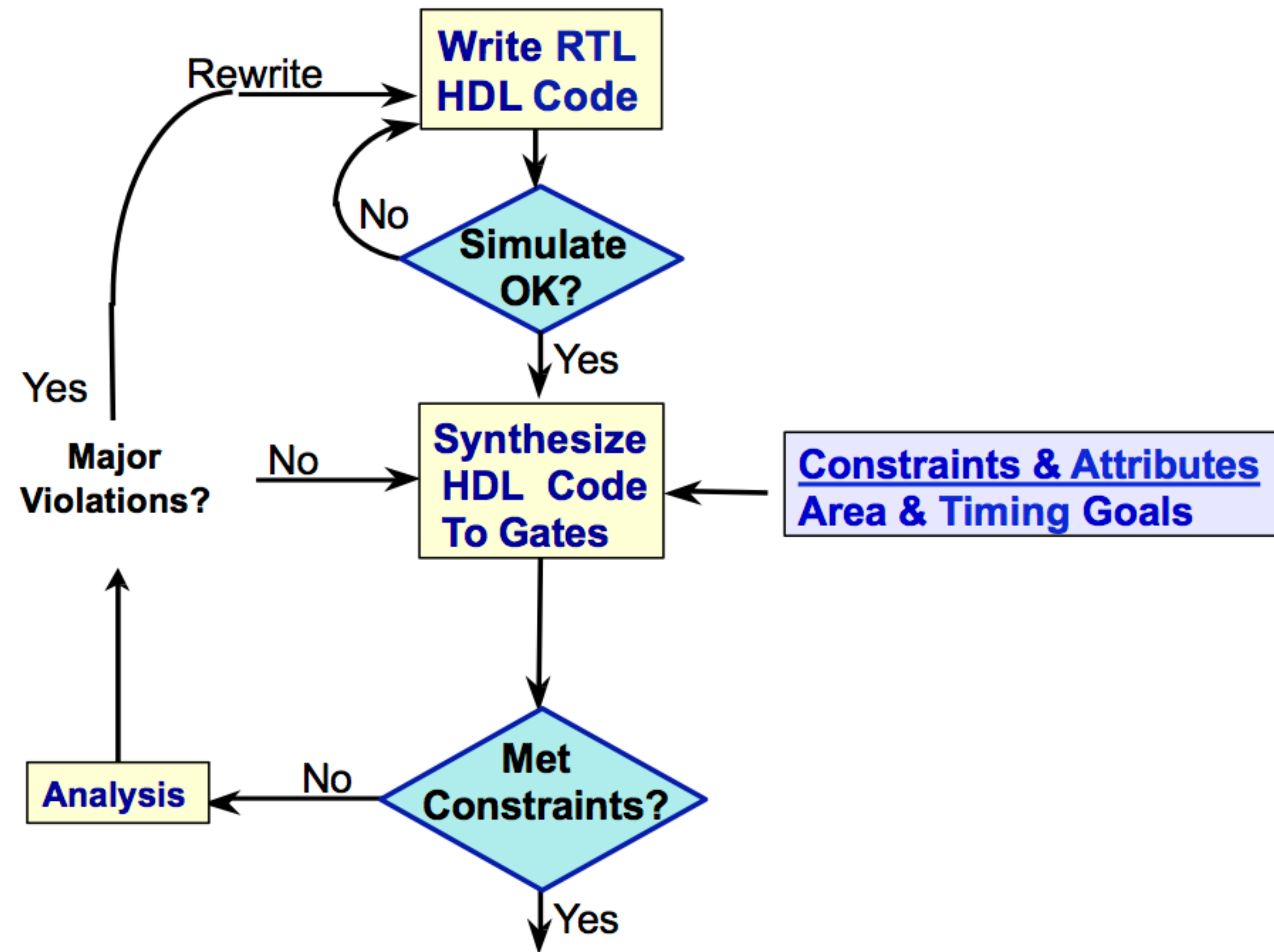
面积约束



ASIC设计流程



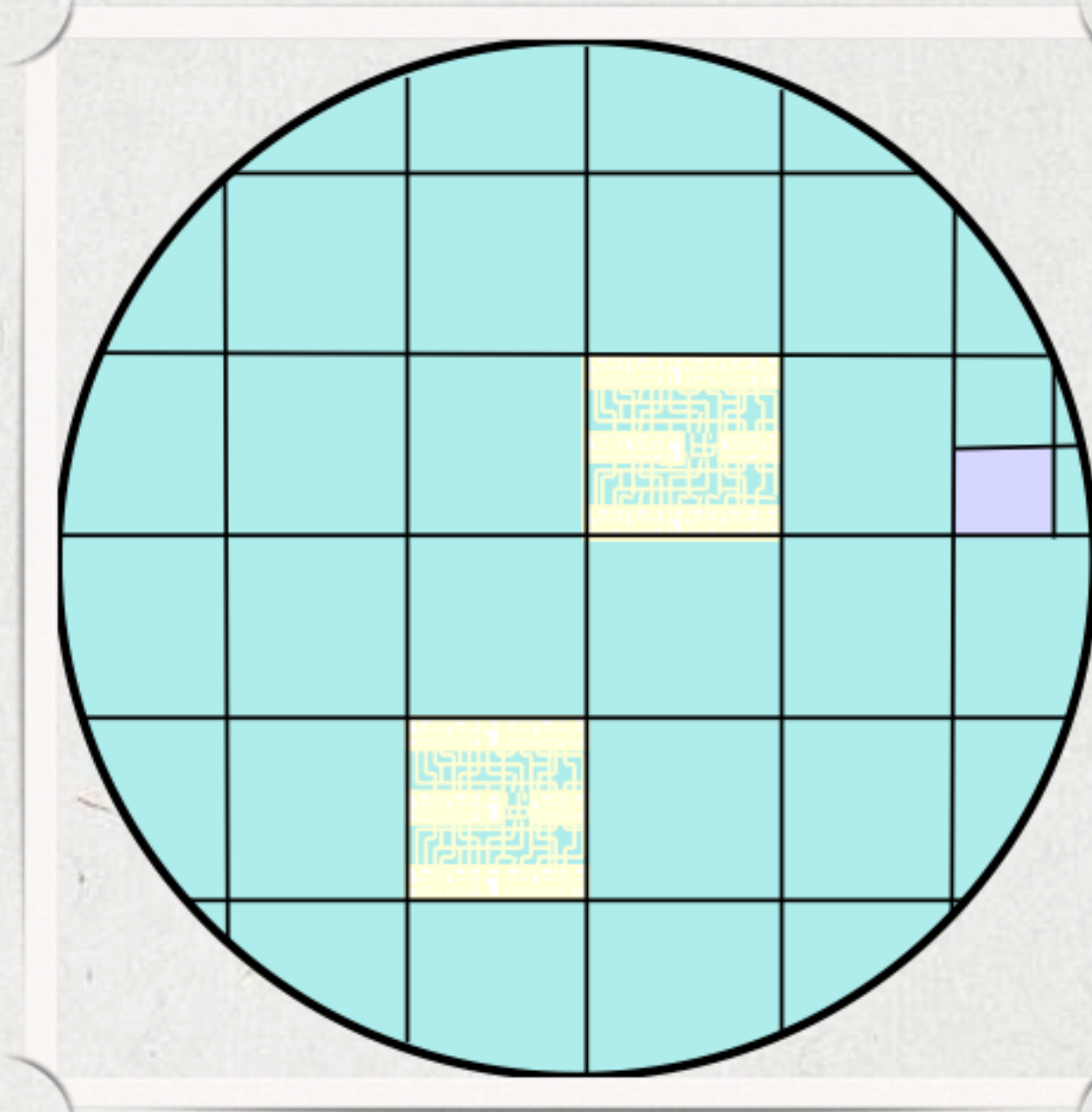
施加约束



定义面积

```
dc_shell-t> current design PRGRM_CNT_TOP  
dc_shell-t> set_max_area 100
```

- Units are those of target library,
defined by the vendor:
 - 2-input-NAND-gate
 - transistors
 - square mils



上面的例子给 PRGRM_CNT_TOP 的设计施加了一个最大面积 100 单位的约束。100 的具体单位是由 Foundry 规定的，定义这个单位有三种可能的标准：

- 第一种是将一个二输入与非门的大小作为单位 1；
- 第二种是以晶体管的数目规定单位；
- 第三种则是根据实际的面积(平方微米等等)。

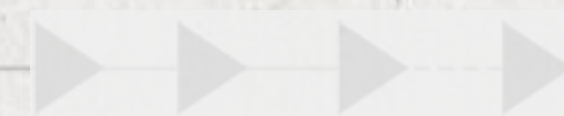
可以通过下面的一个小技巧得到

如果是 1

如果是 4

如果是其他的值

- 即先综合一个二输入与非门，用 report_area 看他的面积是多少。
- 则是按照第一种标准定义的；
- 则是第二种标准；
- 则为第三种标准。

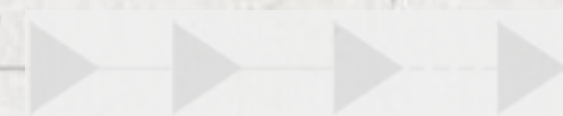


时序路径约束

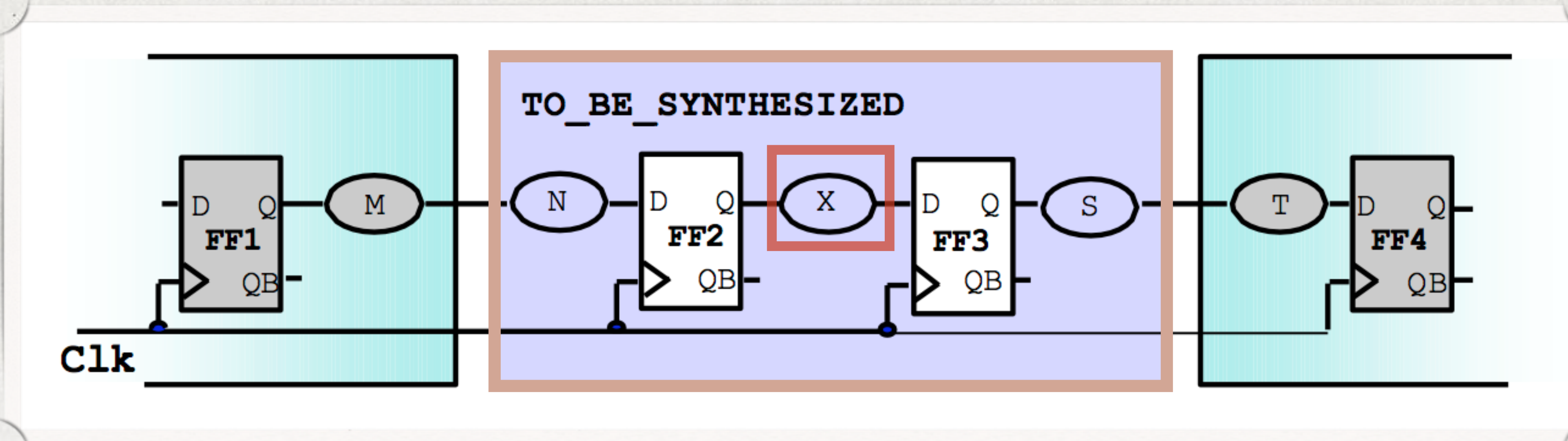
我们要达到的目标是——约束电路中所有的时序路径

这些时序路径可以分为4类：

- 输入到寄存器的路径
- 寄存器到寄存器之间的路径
- 寄存器到输出的路径
- 输入直接到输出的路径



同步设计的时序特点和目标

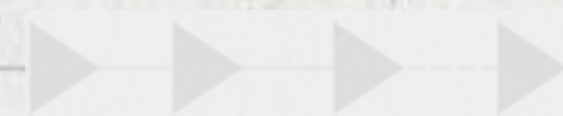


对于图中触发器到触发器的路径，我们应该如何约束？

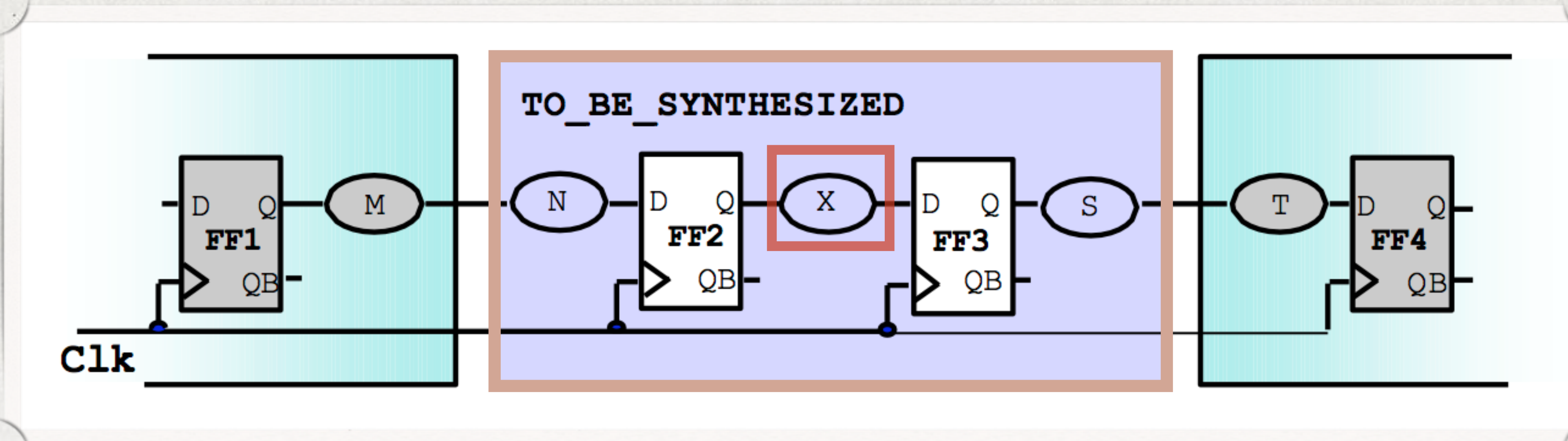
Example:

Clock Period = 10ns

Setup = 1ns



同步设计的时序特点和目标

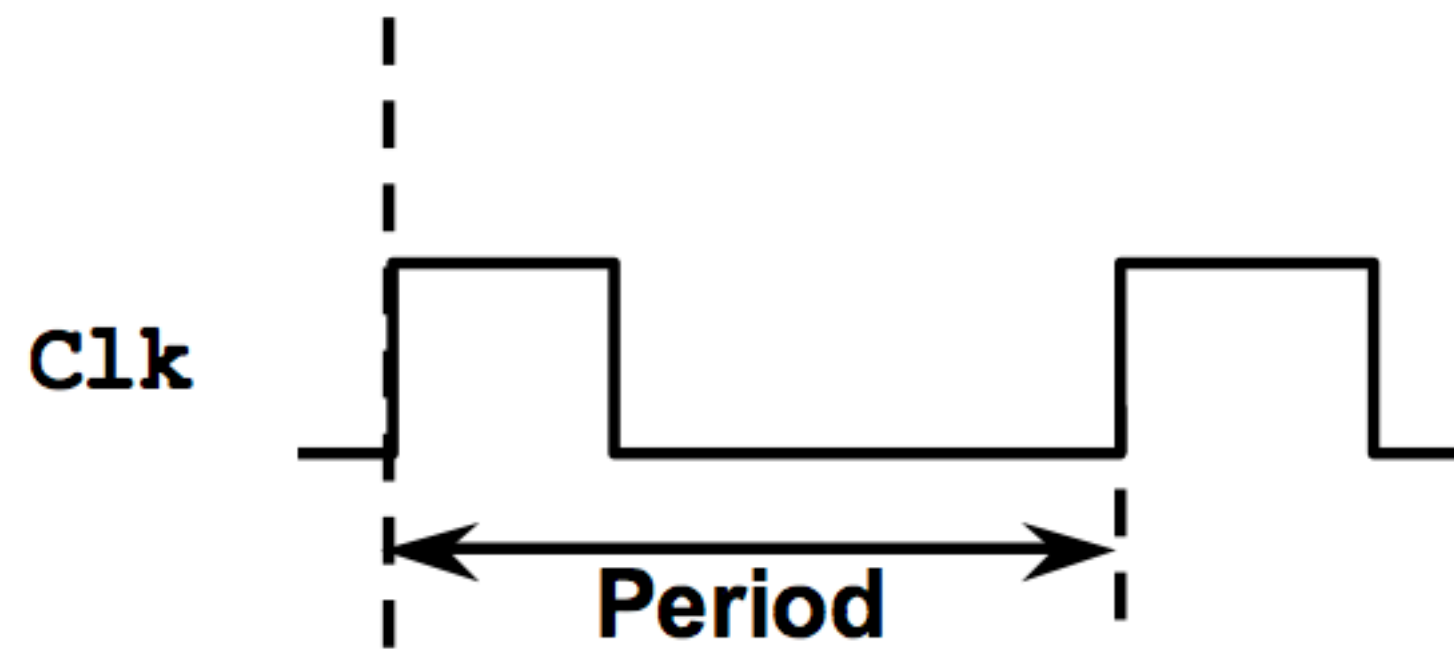


- 在电路综合的过程中，所有时序电路以及组合电路的优化都是以时钟为基准来计算路径延迟的，因此，一般都要在综合的时候指定时钟，作为估计路径延迟的基准。

定义时钟

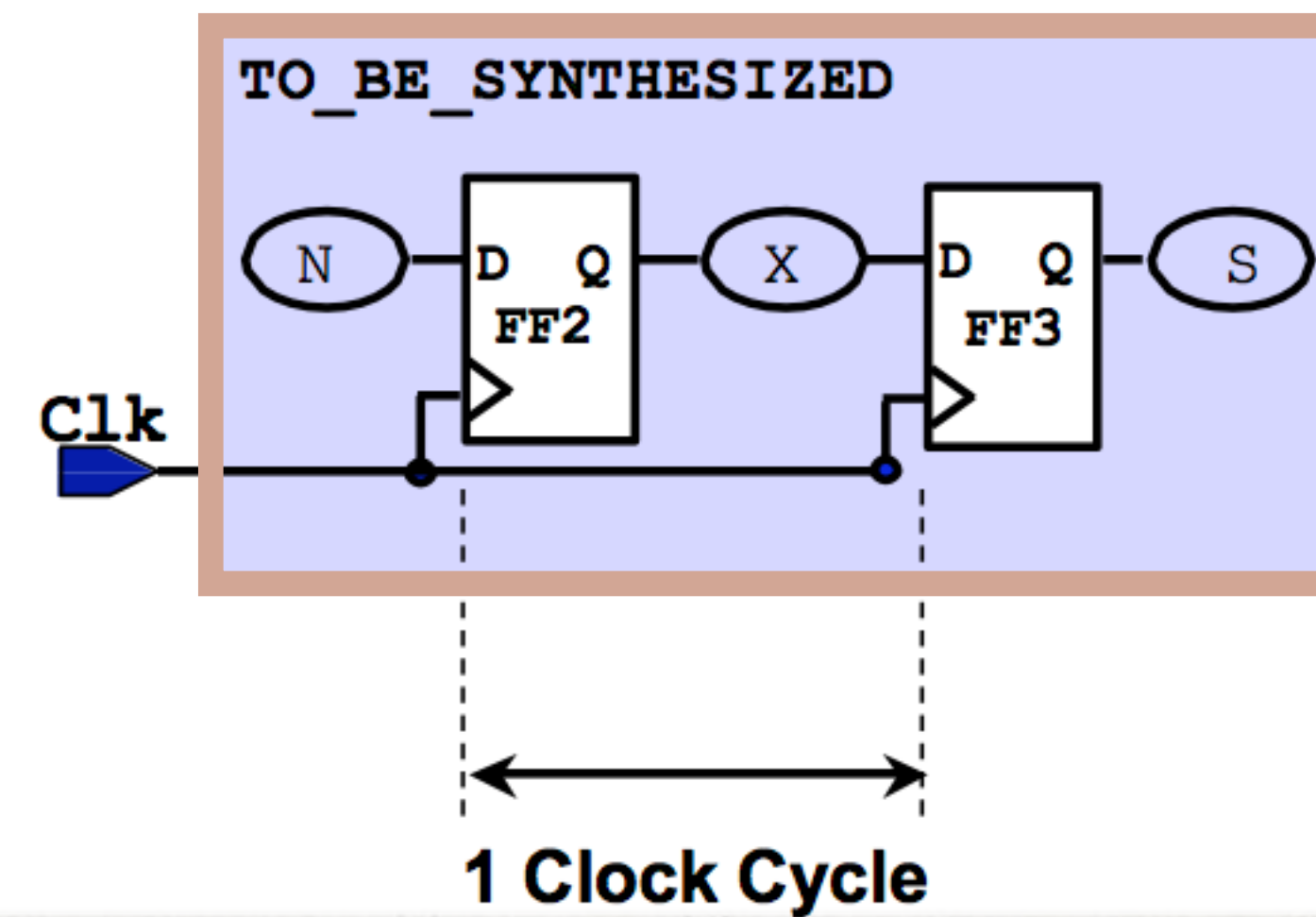
用户必须要定义的值:

- Clock Source (port or pin)
- Clock Period



用户可能需要定义的值:

- Duty Cycle
- Offset/Skew
- Clock Name




```
dc_shell-t> create clock -period 10 [get_ports Clk]
dc_shell-t> set_dont_touch_network [get_clocks Clk]
```

对所有定义的时钟网络设置为 dont_touch，即综合的时候不对 Clk 信号优化。

- 如果不加这句，DC 会根据 Clk 的负载自动对他产生 Buffer，而在实际的电路设计中，时钟树 (Clock Tree)的综合有自己特别的方法，它需要考虑到实际布线后的物理信息，所以 DC 不需要在这里对它进行处理，就算处理了也不会符合要求。

布局布线

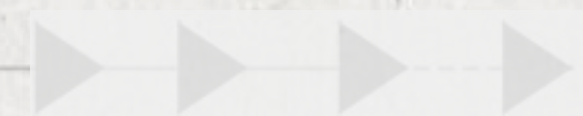


重新做一个正确的网络

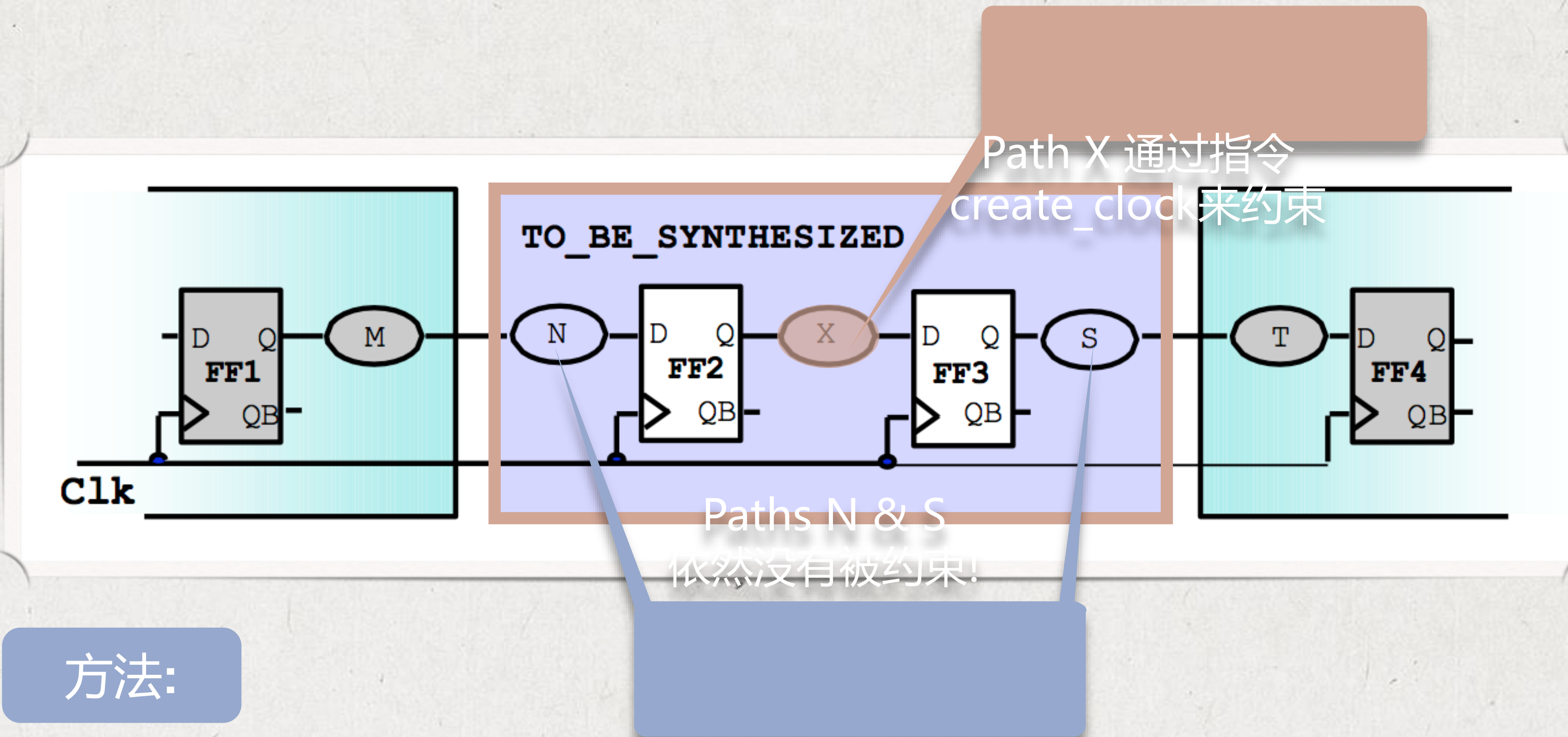

```
dc_shell-t> create_clock -period 10 [get_ports Clk]
dc_shell-t> set_dont_touch_network [get_clocks Clk]
```

对所有定义的时钟网络设置为 dont touch, 即综合的时候不对 Clk 信号优化。

- 如果不加这句, DC 会根据 Clk 的负载自动对他产生 Buffer, 而在实际的电路设计中, 时钟树 (Clock Tree) 的综合有自己特别的方法, 它需要考虑到实际布线后的物理信息, 所以 DC 不需要在这里对它进行处理, 就算处理了也不会符合要求。

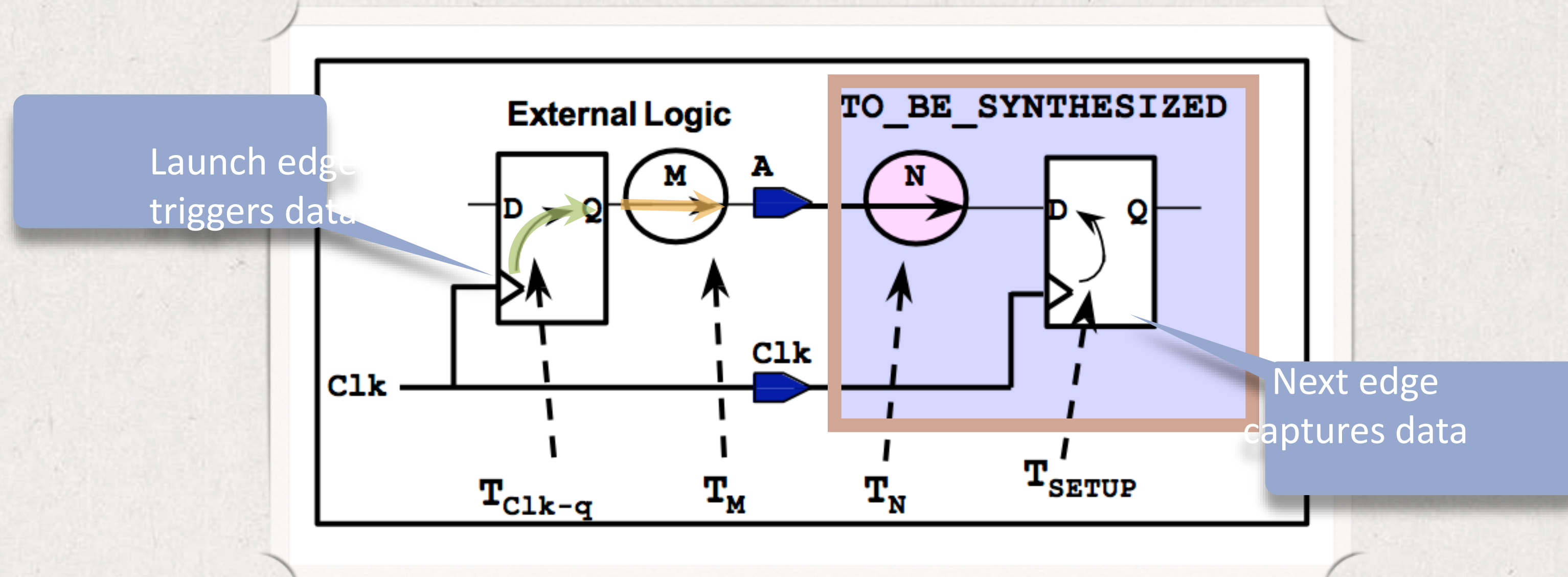


定义时钟



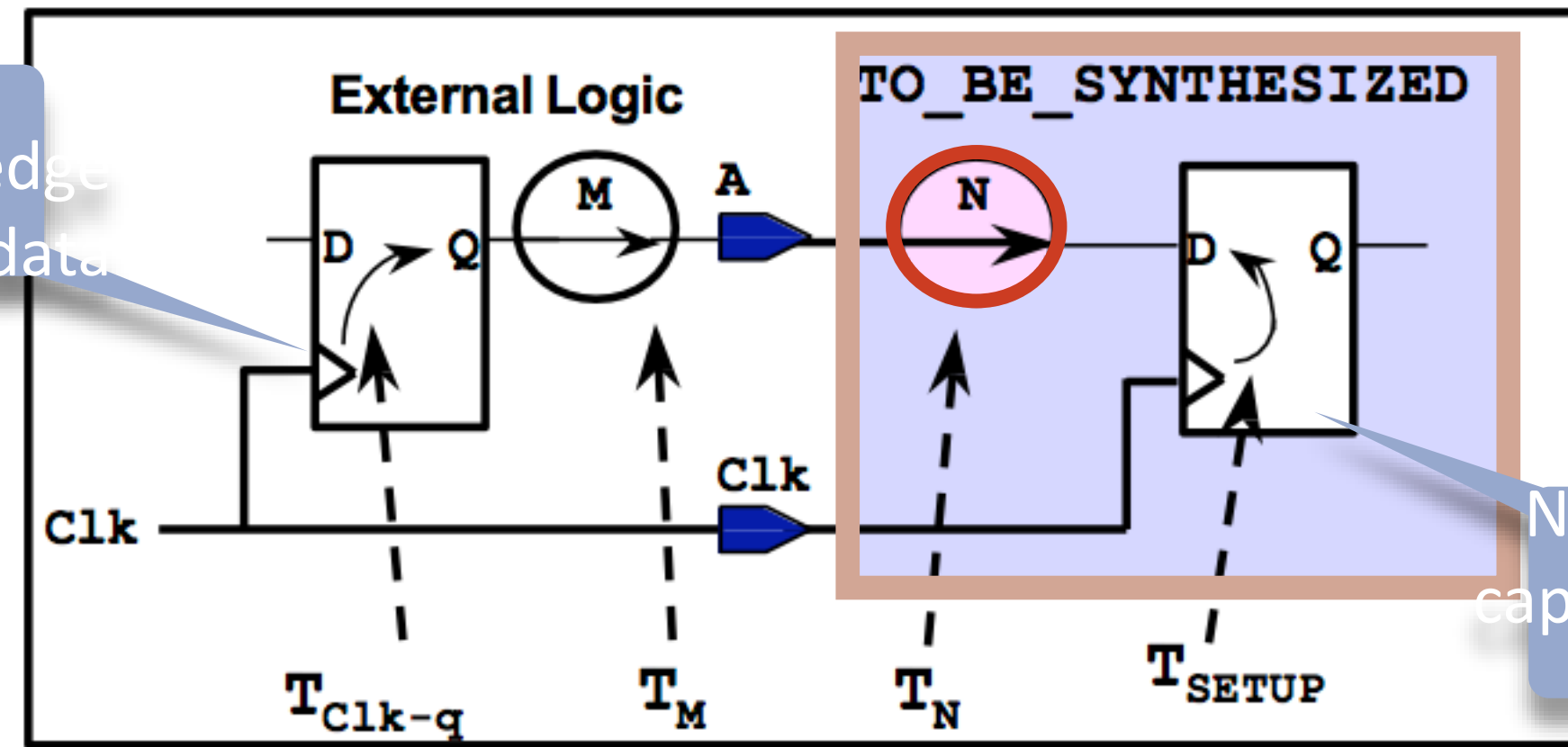
- 定义时钟clock
- 定义 I/O 相对clock的时序

定义输入延迟



- 我们所定义的输入延迟是指被综合模块外的寄存器触发的信号在到达被综合模块之前经过的延迟，在图中就是外围触发器的 clk-q 的延迟加上 M 电路延迟。

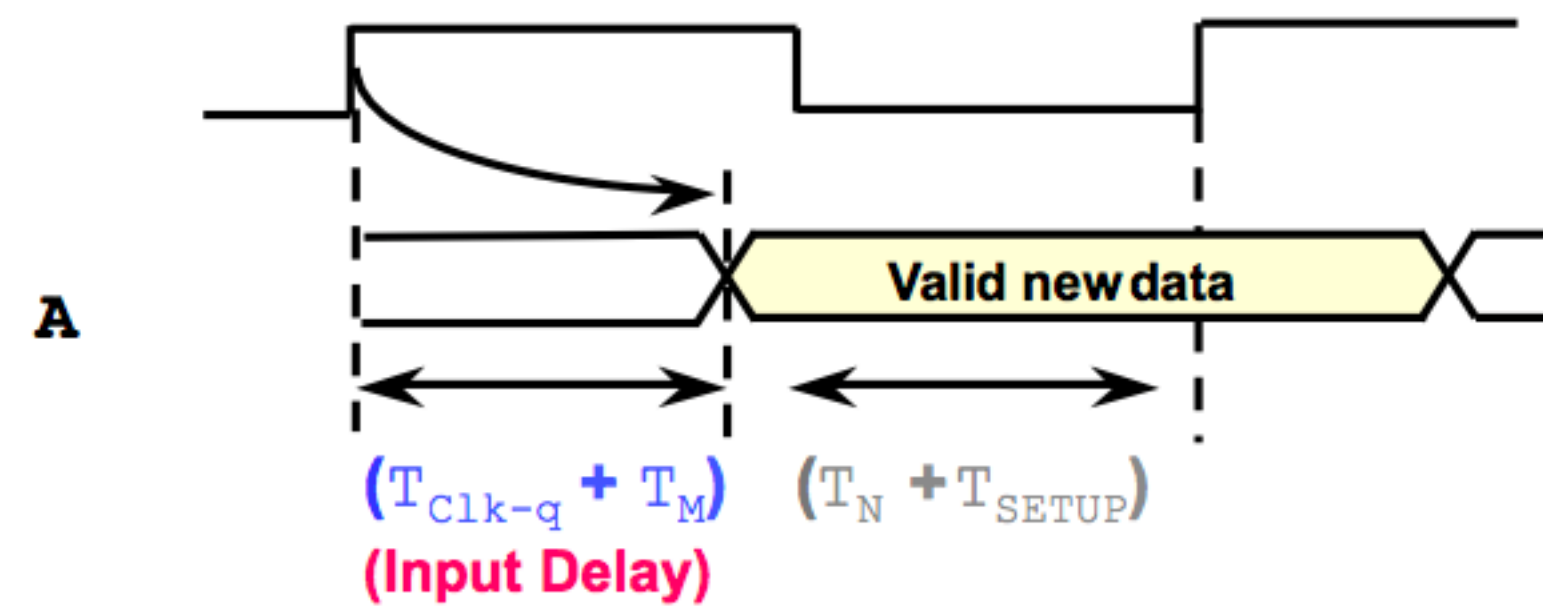
Launch edge
triggers data



Next edge
captures data

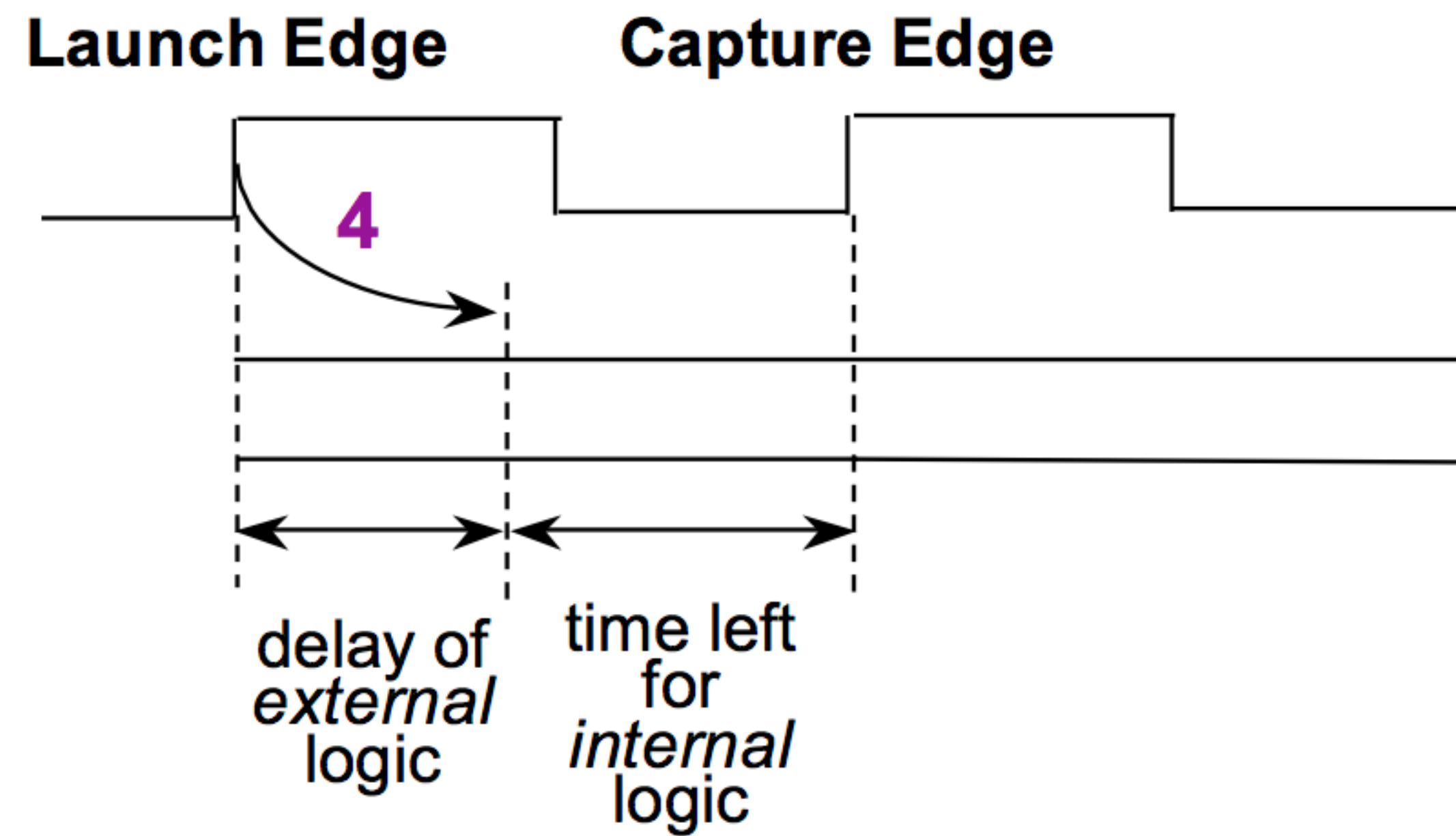
假如时钟周期是 20ns，输入延时是 4ns，内部触发器的建立时间为 1.0ns

- 就可以推断出要使电路正常工作，N 电路的延时最大不能超过 $20 - 4 - 1.0 = 15.0\text{ns}$ 。

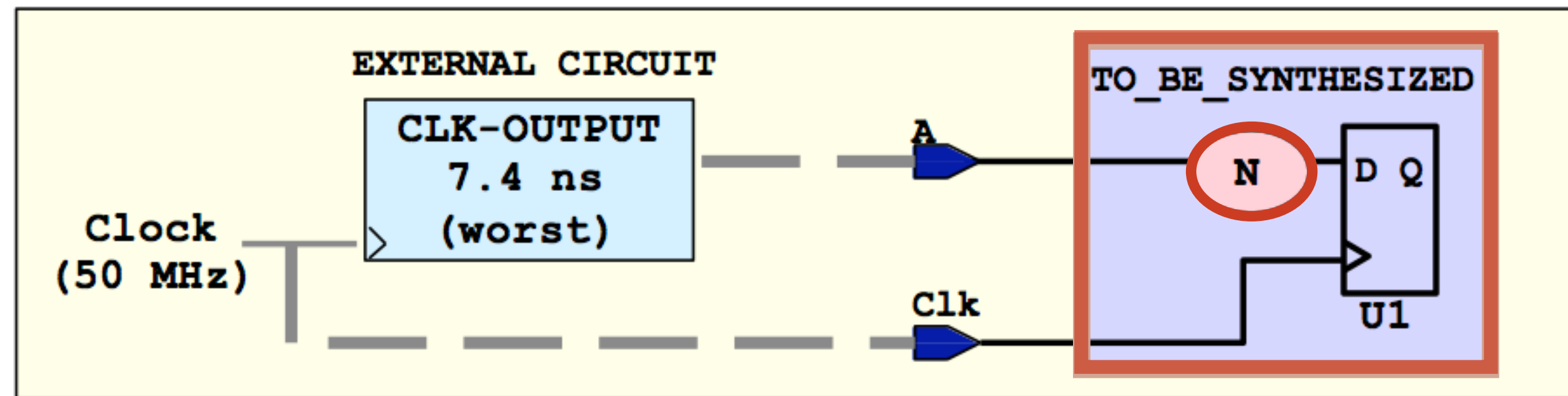


定义输入延迟

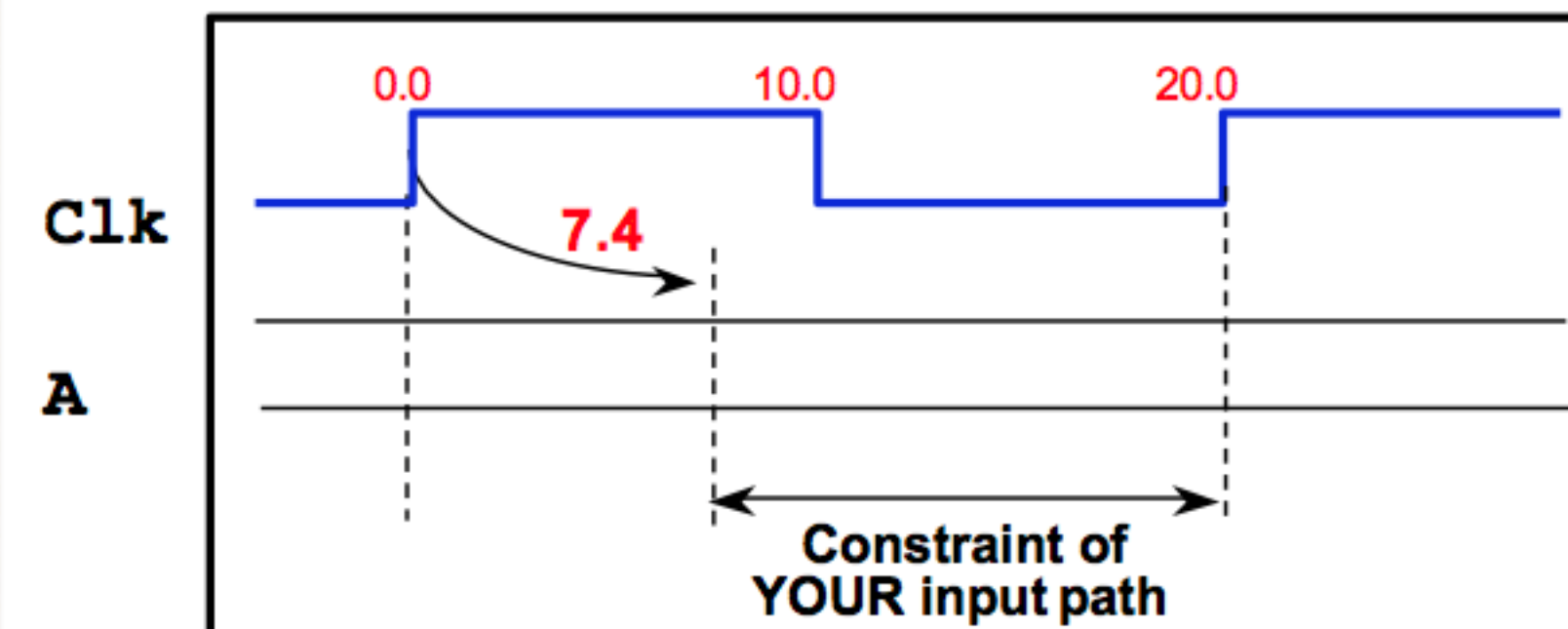
```
dc_shell-t> set_input_delay -max 4 -clock Clk [get_ports A]
```



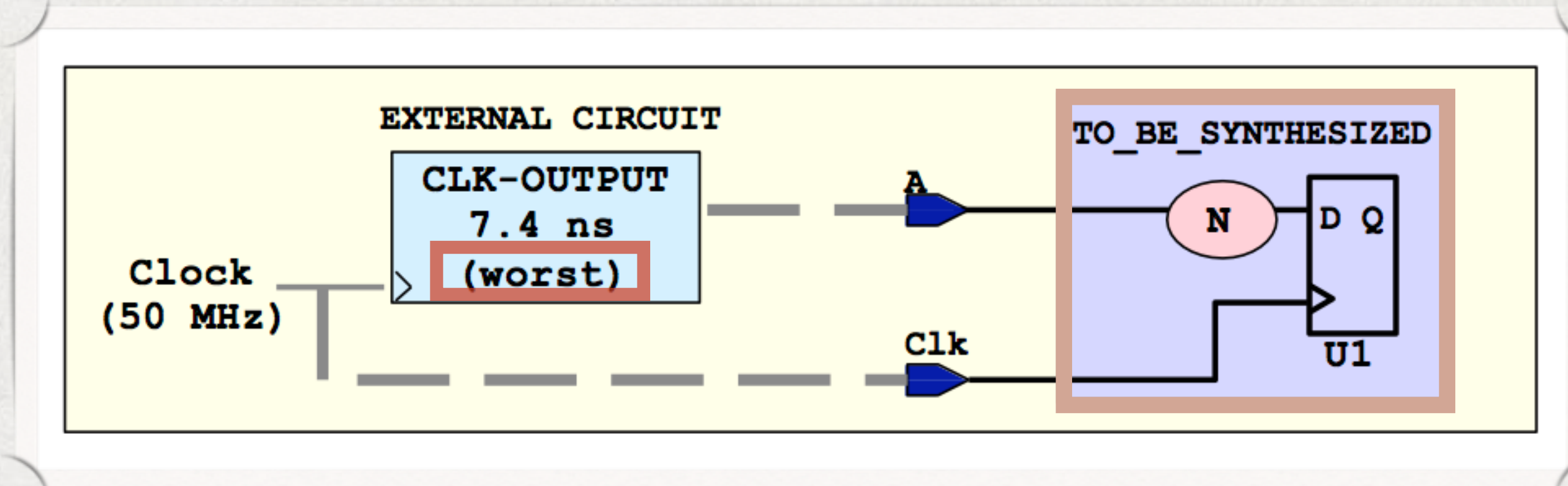
set_input_delay:Exercise



- create_clock _____
- set_dont_touch_network _____
- set_input_delay _____



定义输入延迟

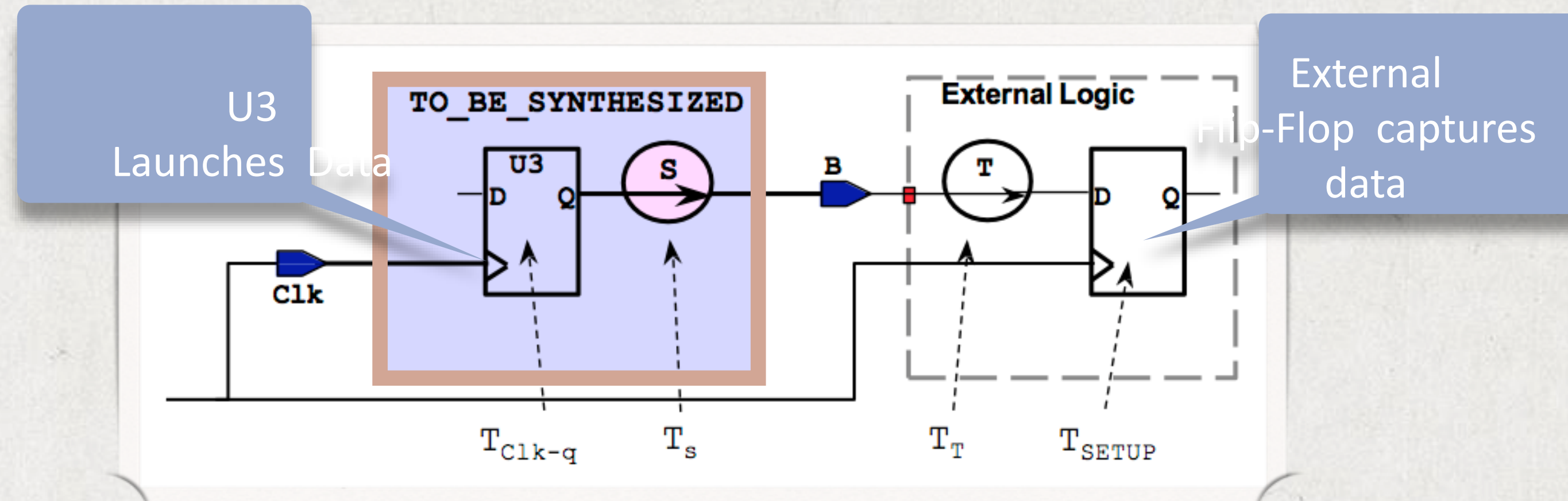


```
create_clock -period 20 [get_ports CLK]
```

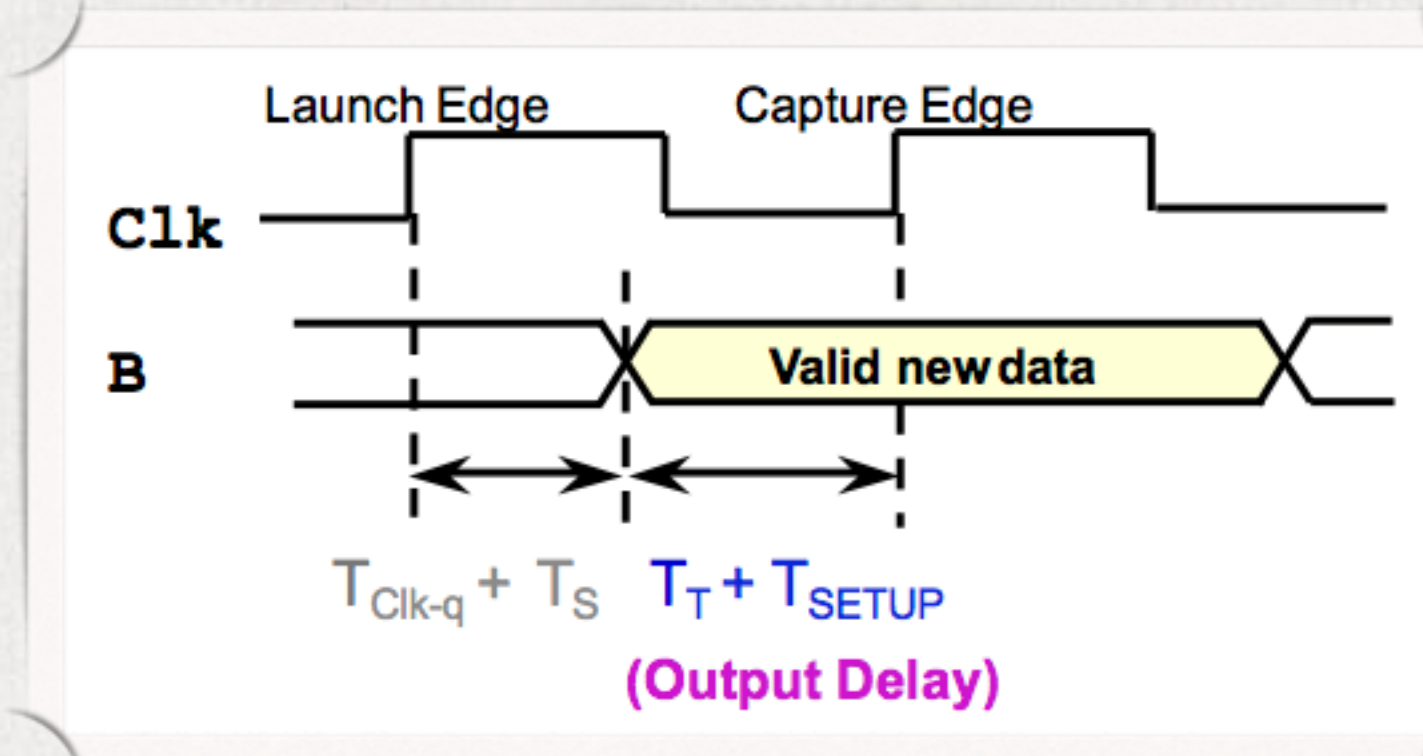
```
set_dont_touch_network [get_clocks CLK]
```

```
set_input_delay -max 7.4 -clock Clk [get_ports A]
```

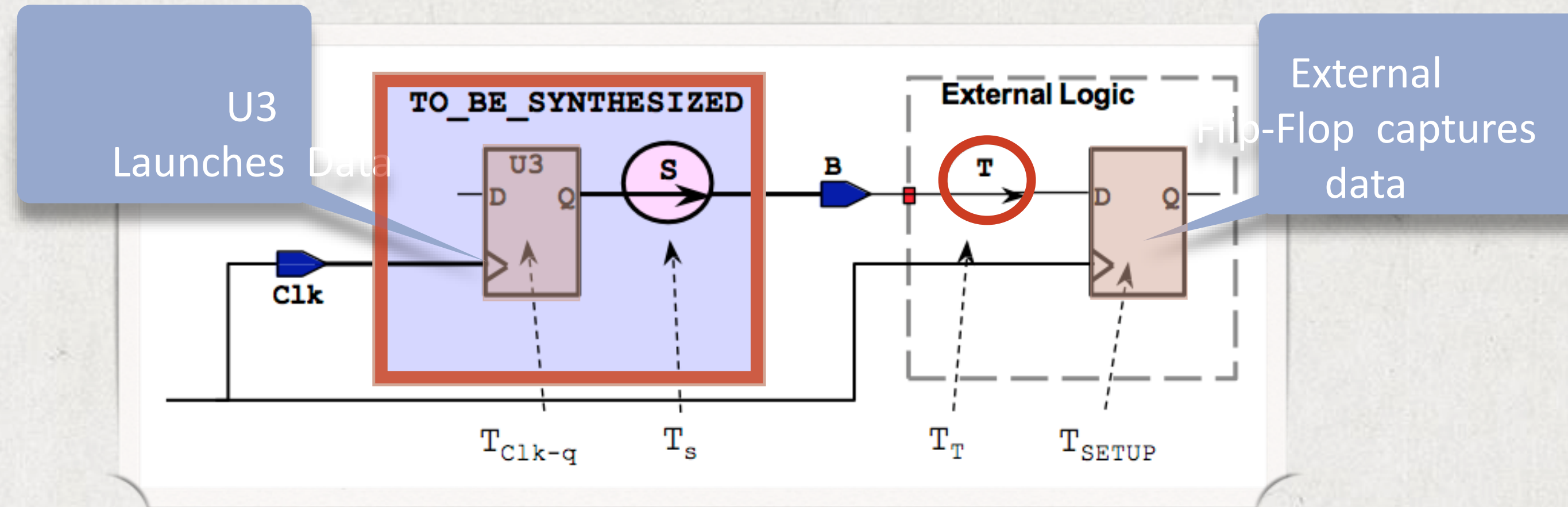

定义输出延迟



为了约束S的path，需要提供哪些信息？



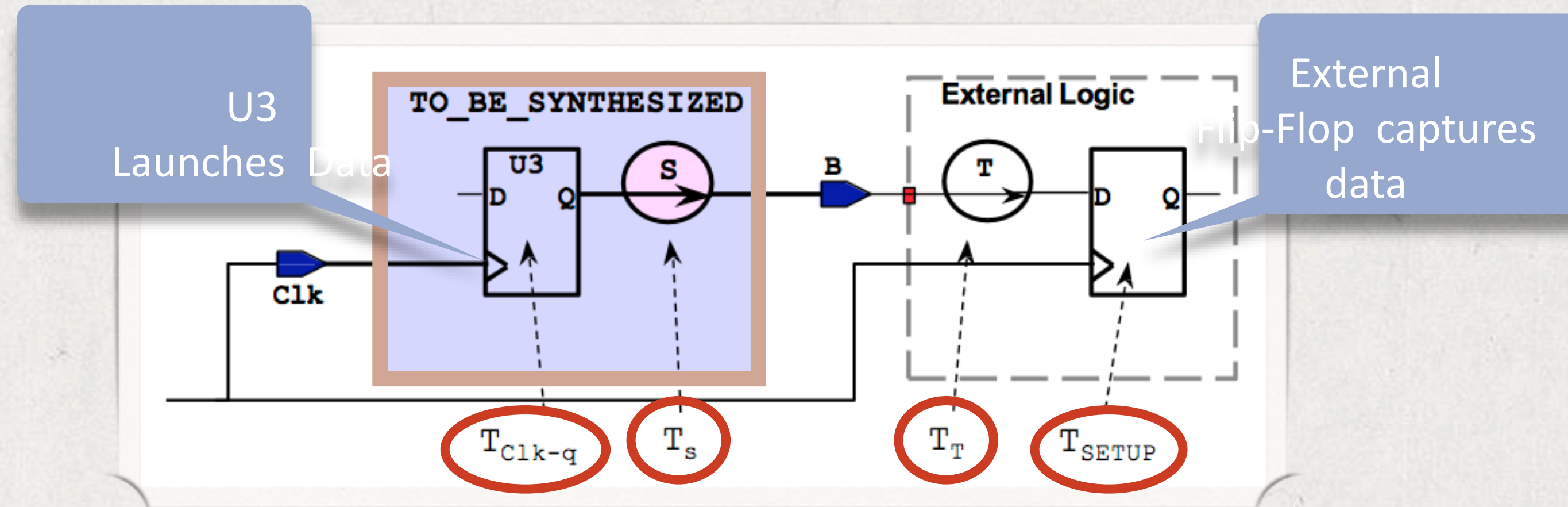
定义输出延迟



What information must you provide to constrain the output paths?

- 信号在被综合模块的触发器 U3 里触发，被外围的一个触发器接收。对外围电路而言，它有一个T 电路延时和外围触发器的建立时间。当确定了他们的延时之后，被综合模块内部的输出路径延时范围也就确定下来了。

定义输出延迟



What information must you provide to constrain the output paths?

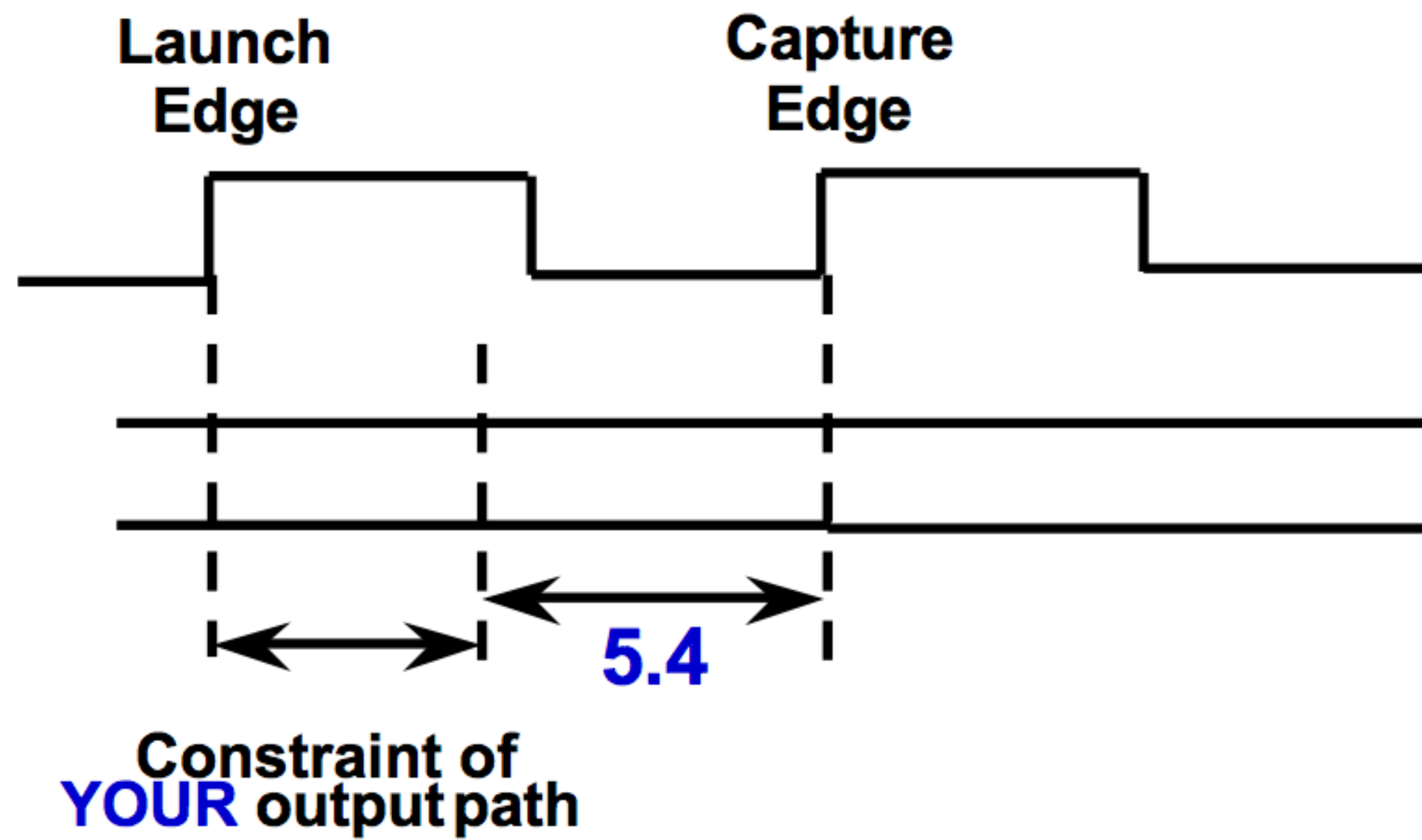
- 假如，时钟周期 20ns，输出延时 5.4ns，U3 触发器的 clk-q 延时为 1.0ns。

输入路径 S 的最大延时:

$$20 - 5.4 - 1.0 = 13.6\text{ns}$$

定义输出延迟

```
dc_shell-t> set_output_delay -max 5.4 -clock Clk [get_ports B]
```



rule constraints

set_max_transition

data transition time

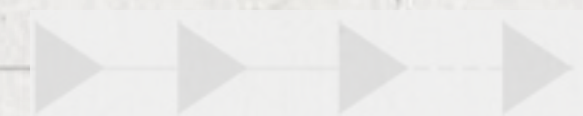
clock transition time

小

- 是约束design中的信号、端口、net最大transition不能超过这个值，当然是越小越严苛了，net的transition time取决于net的负载(fanout)，负载越大，transition time越大。

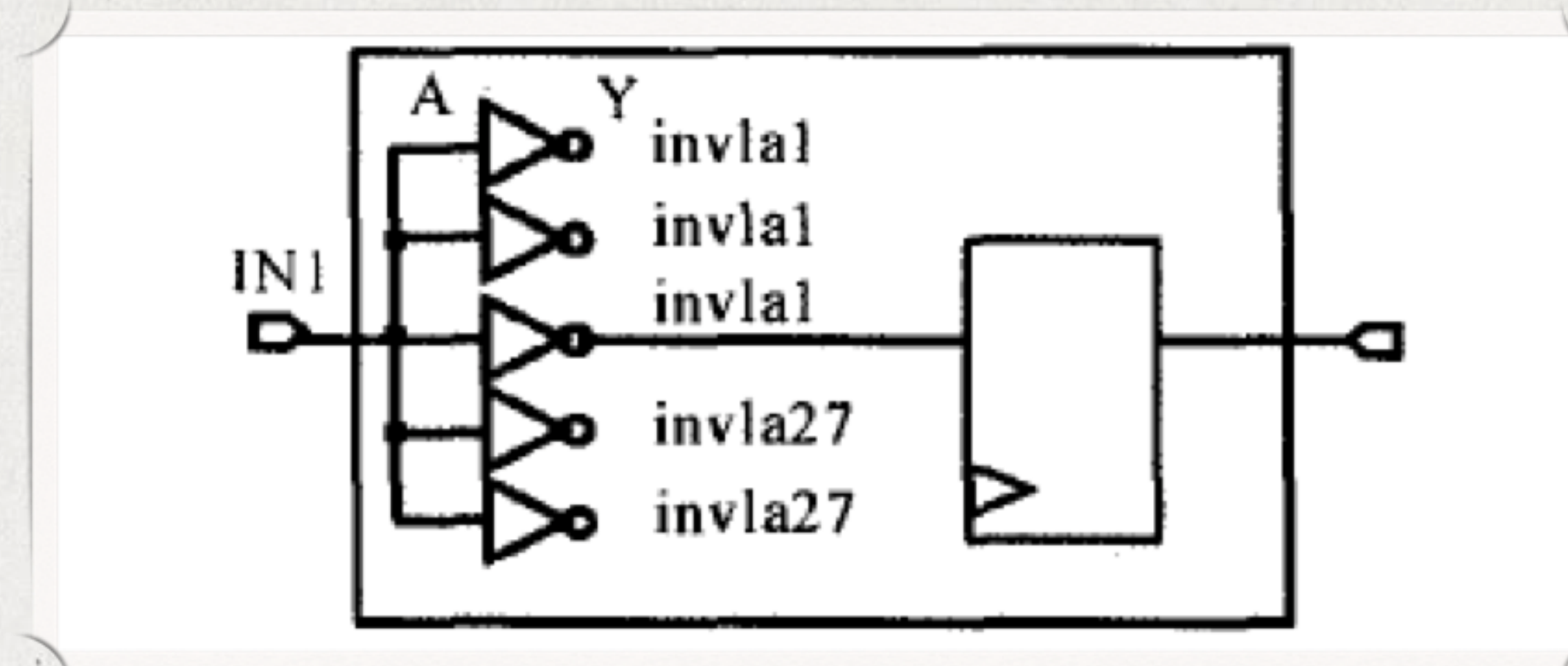
set_max_fanout

- 对design,net,output port进行操作，设定的不是具体的电容值。扇出负载值是用来表示单元输入引脚相对负载的数目，它并不表示真正的电容负载，而是个无量纲的数字。



set_max_capacitance

- 基于工艺库的信息进行设定。



- 以上三个由工艺厂商提供，但是我们可以约束的更紧一些。
- 数字IC设计中max_cap和max_tran这类逻辑DRC或者说时序DRC是在设计中必须修复的问题，否则无法流片。

TCL脚本

```
sh date      //显示开始时间
remove_design -designs //移除DC中原有的设计
//下面是库的设置，对应图形界面操作的2
#####
#set library      #
#####
set search_path [list *****]
set target_library { tt.db }
set link_library { * tt.db }
set symbol_library { tt.sdb }
```


//下面是屏蔽一些warning信息，DC在综合时遇到这些warning时就把它们忽略，不会报告这些信息，VER-130，VER-129等是不同warning信息的编码，具体含义可以查看帮助

```
#####
```

```
#void warning Info    #
```

```
#####
```

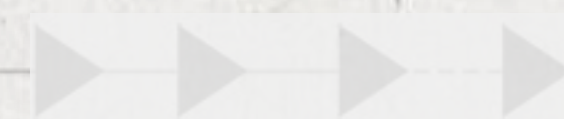
```
suppress_message VER-130
```

```
suppress_message VER-129
```

```
suppress_message VER-318
```

```
suppress_message ELAB-311
```

```
suppress_message VER-936
```



//读入example1.v文件，对应于图形界面的3
#####

#read&link&Check design#

#####

read_file -format verilog ~/example1.v

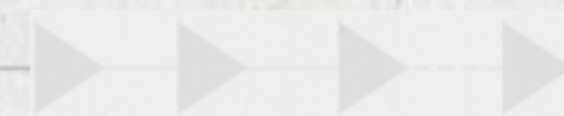
#analyze -format verilog ~/example1.v

#elaborate EXAMPLE1

current_design EXAMPLE1 //把EXAMPLE1指定为当前设计的顶层模块

uniquify

check_design



//设置一些变量

#####

define IO port name

#####

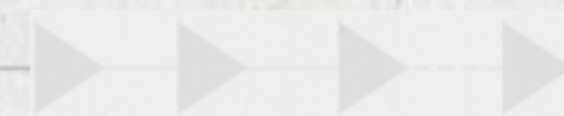
set clk [get_ports clk] //设置变量clk的值是[get_ports clk]，在下面的代码中若出现\$clk字样，则表示引用该变量的值，即用[get_ports clk]代替\$clk。

set rst_n [get_ports rst_n]

set general_inputs [list a b c]

set outputs [get_ports o]

get ports



//设置约束条件，对应于图形界面的4

#####

set_constraints

#####

//设置时钟约束，对应于图形界面的4.1


#1 set constraints for clock signals

create_clock -n clock \$clk -period 20 -waveform {0 10} //创建一个周期为20ns，占空比为1的时钟

set_dont_touch_network [get_clocks clock]

set_drive 0 \$clk //设置时钟端口的驱动为无穷大

set_ideal_network [get_ports clk] //设置时钟端为理想网络



//设置复位信号约束，对应于图形界面的4.2

#2 set constraints for reset signals

set_dont_touch_network \$rst_n

set_drive 0 \$rst_n

set_ideal_network [get_ports rst_n]





//设置输入延时，对应图形界面的4.3

#3 set input delay

set_input_delay -clock clock 8

\$general_inputs

//设置输出延时，对应图形界面的4.4

#4 set output delay

set_output_delay -clock clock 8

\$outputs





//设置面积约束和设计约束，对应图形界面的4.5

#5 set design rule constraints

set_max_fanout 4 \$general_inputs

set_max_transition 0.5 [get_designs

"EXAMPLE1"]

#6 set area constraint

set_max_area 0



在时序满足的条件下，尽可能把电路面积做小一些。

