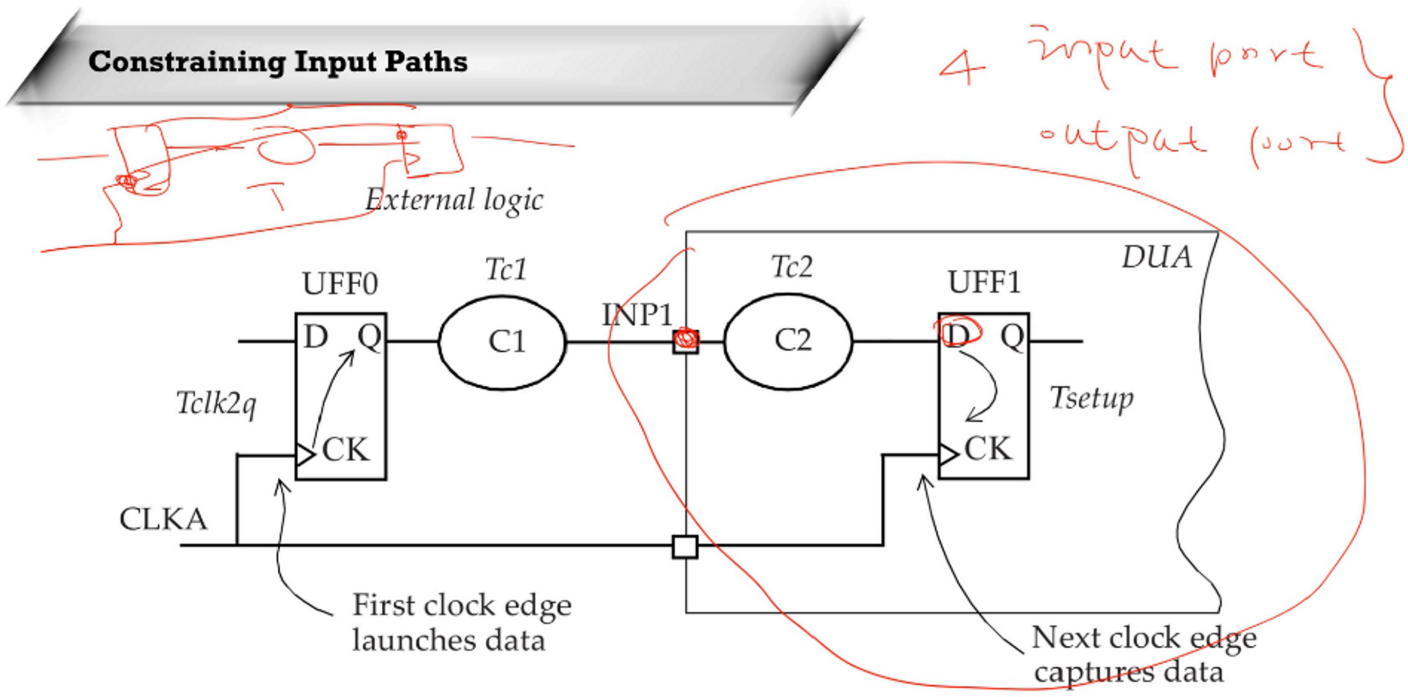
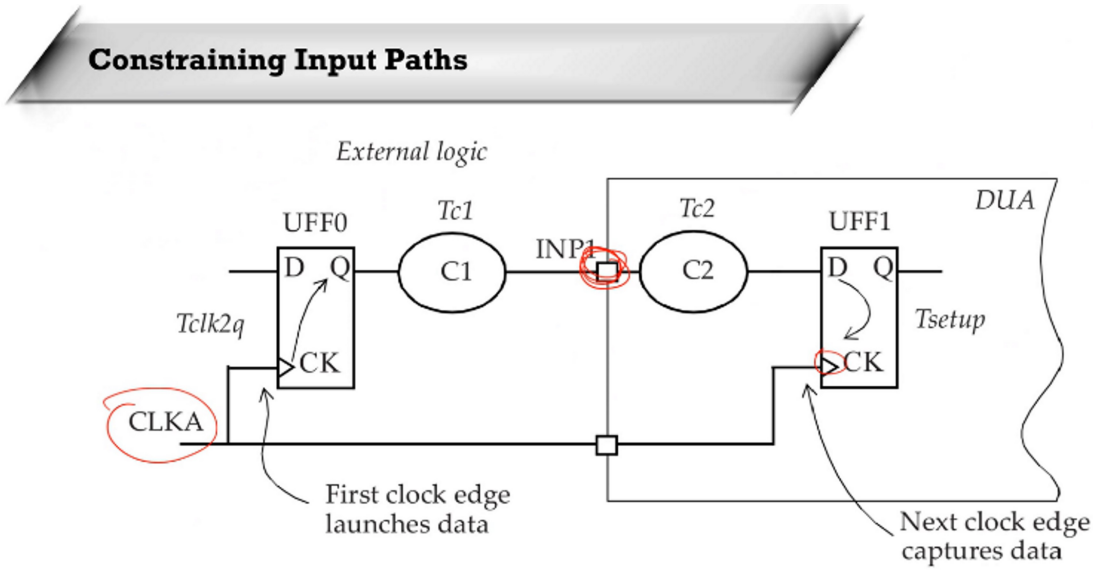
SDC constraints II

2022年6月9日　14:06

STA environment - SDC constraints

Input_delay：考虑input delay，从而约束input port到D pin的timing path
这里用到的还是同一个时钟源，所以不需要设置虚拟时钟






```
set Tclk2q 0.9
set Tc1 0.6
set_input_delay -clock CLKA -max [expr Tclk2q + Tc1] [get_ports INP1]
```

max：input delay的最大值
min：input delay的最小值

Constraining Input Paths

```
create_clock -period 15 -waveform {5 12}
[get_ports CLKP]
set_input_delay -clock CLKP -max 6.7
[get_ports INPA]
set_input_delay -clock CLKP -min 3.0
[get_ports INPA]
```

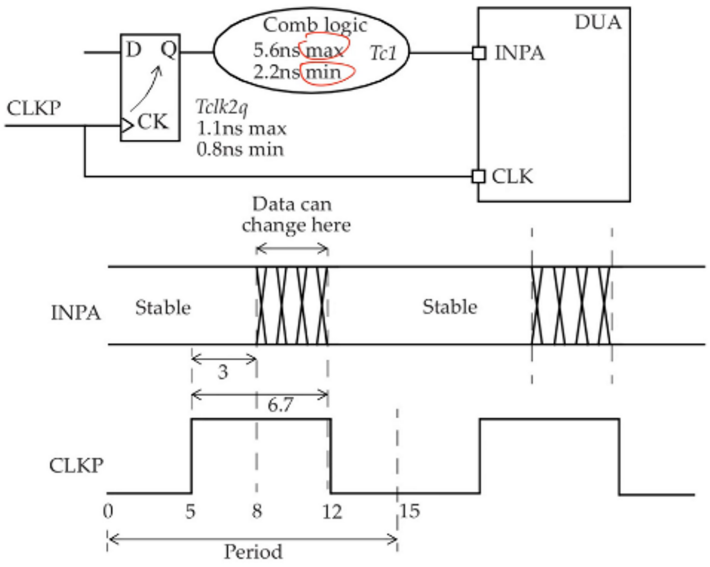


Figure 7-22 *Max and min delays on input port.*

The waveform on INPA shows the window in which
the data arrives at the design input and when it is expected to be stable.

时钟周期是15，考虑max delay对于setup time的限制，max delay = 6.7，那么剩下可用的时间为8.3，对于INPA内部，剩下可用的8.3之前的时间（考虑max delay更大，剩下可用的时间比8.3更小），数据必须稳定下来

output_delay：考虑output delay，从而约束CK pin到output port的timing path

Constraining Output Paths

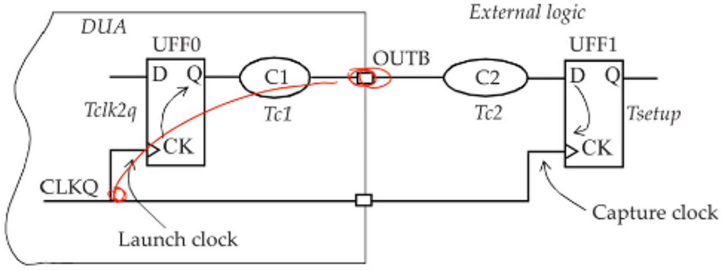


Figure 7-23 *Output port timing path for example A.*

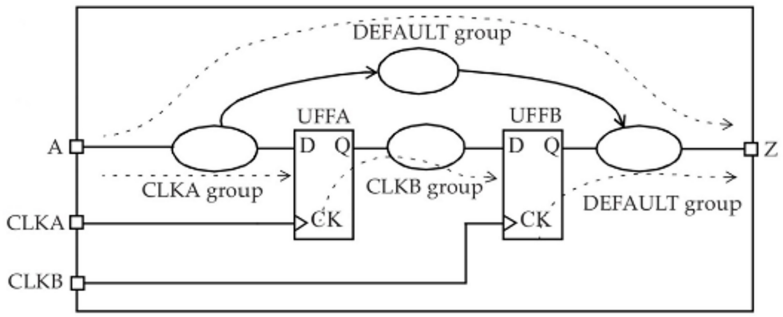
```
set Tc2 3.9
set Tsetup 1.1
set_output_delay -clock CLKQ -max [expr Tc2 + Tsetup] [get_ports OUTB]
```

Timing Path Groups

Timing Path Groups

Timing paths are sorted into path groups by the clock associated with the **endpoint** of the path. Thus, each clock has a set of paths associated with it.

There is also a default path group that includes all non-clocked (asynchronous) paths.



为了定义Timing path属于哪个时钟组，只需要看end point对应的时钟

比方说，这里CK pin to D pin，end point是D pin，相关的时钟是CLKB，所以CK pin to D pin是CLKB组

Timing path有4种：

- Reg to Reg (Normal)
- Reg to output port (set_output_delay)
- Input port to reg (set_input_delay)
- Input port to output port (both set_input_delay && set_output_delay)

为了对所有timing path进行约束，另外要设置drive, driving_cell, input_transition, load

Modeling of External Attributes

While create_clock, set_input_delay and set_output_delay are enough to constrain all paths in a design for performing timing analysis, these are not enough to obtain accurate timing for the IO pins of the block.

The following attributes are also required to accurately model the environment of a design.

Modeling of External Attributes

For inputs, one needs to specify the slew at the input. This information can be provided using:

- set_drive
- set_driving_cell
- set_input_transition

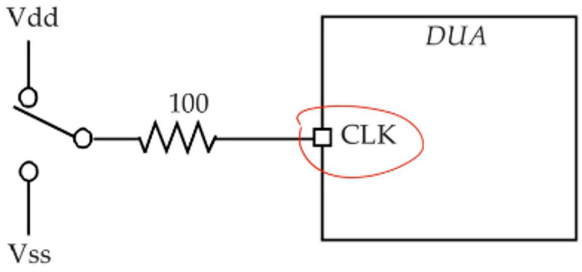
For outputs, one needs to specify the capacitive load seen by the output pin. This is specified by using the following specification:

- set_load

set_drive resistance_value
drive越小，驱动强度越大，0是理想值

Modeling of External Attributes

The set_drive explicitly specifies a value for the drive resistance at the input pin of the DUA. The smaller the drive value, the higher the drive strength.
A resistance value of 0 implies an infinite drive strength.



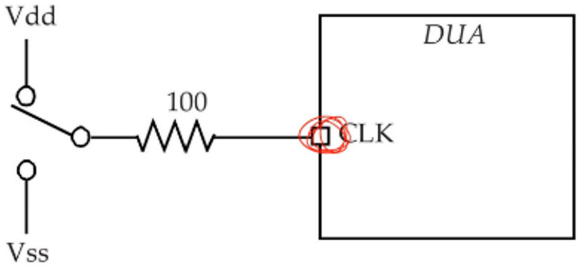

```
set_drive 100 UCLK
# Specifies a drive resistance of 100 on input UCLK.
```

```
# Rise drive is different from fall drive:
set_drive -rise 3 [all_inputs]
set_drive -fall 2 [all_inputs]
```

Modeling of External Attributes

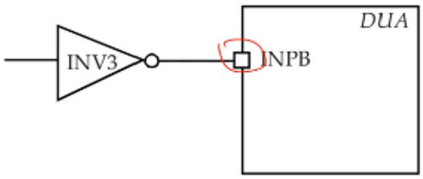
The drive of an input port is used to calculate the transition time at the first cell. The drive value specified is also used to compute the delay from the input port to the first cell in the presence of any RC interconnect.

```
Delay_to_first_gate = (drive * load_on_net) + interconnect_delay
```



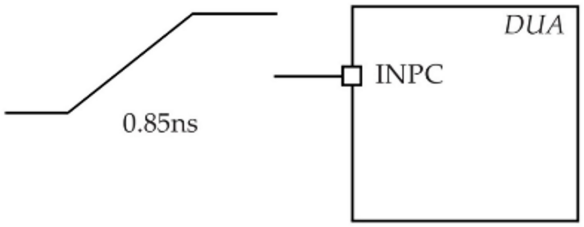
Modeling of External Attributes

The set_driving_cell specification offers a more convenient and accurate approach in describing the drive capability of a port.




```
set_driving_cell -lib_cell INV3  -library slow [get_ports INPB]
# The input INPB is driven by an INV3 cell from library slow.
set_driving_cell -lib_cell INV2 -library tech13g [all_inputs]
# Specifies that the cell INV2 from a library tech13g is the driving cell for all inputs.
set_driving_cell -lib_cell BUFFD4 -library tech90gwc [get_ports {testmode[3]}]
# The input testmode[3] is driven by a BUFFD4 cell from library tech90gwc.
```

Modeling of External Attributes




```
set_input_transition 0.85 [get_ports INPC]
# Specifies an input transition of 850ps on port INPC.
set_input_transition 0.6 [all_inputs]
# Specifies a transition of 600ps on all input ports.
set_input_transition 0.25 [get_ports SD_DIN*]
# Specifies a transition of 250ps on all ports with pattern SD_DIN*.
# Min and max values can optionally be specified using the -min and -max options.
```

第一个cell delay要正确获得，必须设置input_transition

Modeling of External Attributes

- In summary, a slew value at an input is needed to determine the delay of the first cell in the input path.
- In the absence of this specification, an ideal transition value of 0 is assumed, which may not be realistic.

Modeling of External Attributes

The set_load specification places a capacitive load on output ports to model the external load being driven by the output port.

By default, the capacitive load on ports is 0. The load can be specified as an explicit capacitance value or as an input pin capacitance of a cell.

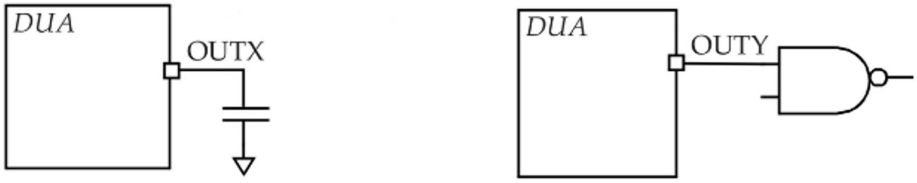


Figure 7-31 *Capacitive load on output port.*

Modeling of External Attributes

```
set_load 5 [get_ports OUTX]
# Places a 5pF load on output port OUTX.
set_load 25 [all_outputs]
# Sets 25pF load capacitance on all outputs.
set_load -pin_load 0.007 [get_ports {shift_write[31]}]
# Place 7fF pin load on the specified output port.
# A load on the net connected to the port can be specified using the -wire_load option.
# If neither -pin_load nor -wire_load option is used, # the default is the -pin_load option.
```

It is important to specify the load on outputs since this value impacts the delay of the cell driving the output. In the absence of such a specification, a load of 0 is assumed which may not be realistic.

```
set_load [get_attribute [get_lib_pins tech_lib/NAND2/A] pin_capacitance] [all_outputs]
```

Design Rule Check
based on lib setting
set_max_transition
set_max_capacitance
set_max_area
set_max_fanout

Design Rule Checks

Two of the frequently used design rules for STA are max transition and max capacitance. These rules check that all ports and pins in the design meet the specified limits for transition time 1 and capacitance.

These limits can be specified using:

- set_max_transition
- set_max_capacitance

```
set_max_transition 0.6 IOBANK
# Sets a limit of 600ps on IOBANK.
set_max_capacitance 0.5 [current_design]
# Max capacitance is set to 0.5pf on all nets in current design.
```

Design Rule Checks

There are other design rule checks that can also be specified for a design.

These are: set_max_fanout (specifies a fanout limit on all pins in design), set_max_area (for a design); however these checks apply for synthesis and not for STA.

定义虚拟时钟：
虚拟时钟没有真实的点(port && pin)

Virtual Clocks

A virtual clock is a clock that exists but is not associated with any pin or port of the design. It is used as a reference in STA analysis to specify input and output delays relative to a clock.

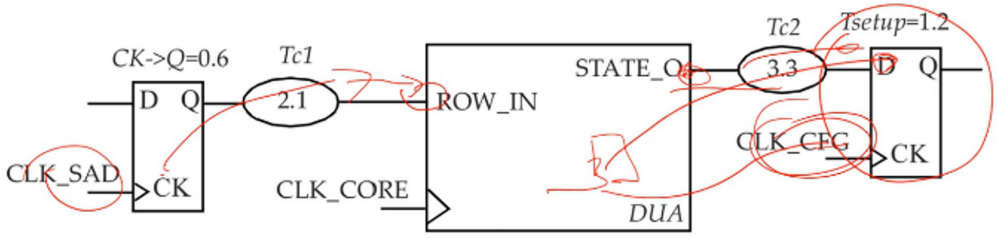


Figure 7-33 *Virtual clocks for CLK_SAD and CLK_CFG.*

Virtual Clock是设置参考的基准线
这里用到了虚拟时钟定义input delay和output delay

Virtual Clocks

```
create_clock -name VIRTUAL_CLK_SAD -period 10 -waveform {2 8}
create_clock -name VIRTUAL_CLK_CFG -period 8 -waveform {0 4}
create_clock -period 10 [get_ports CLK_CORE]
set_input_delay -clock VIRTUAL_CLK_SAD -max 2.7 [get_ports ROW_IN]
set_output_delay -clock VIRTUAL_CLK_CFG -max 4.5 [get_ports STATE_O]
```

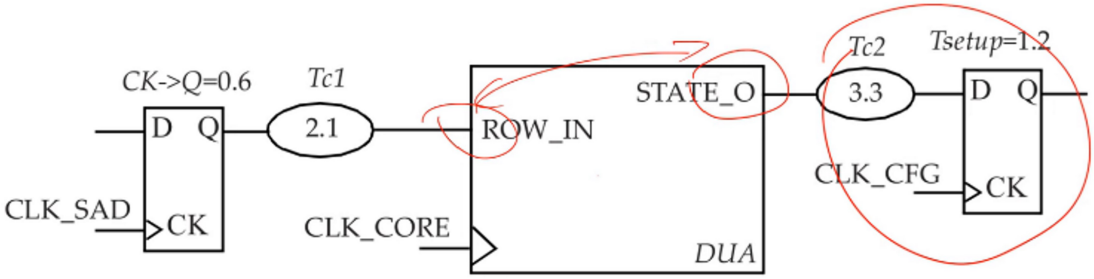


Figure 7-33 *Virtual clocks for CLK_SAD and CLK_CFG.*

7.9 Virtual Clocks

A **virtual clock** is a clock that exists but is not associated with any pin or port of the design. It is used as a reference in STA analysis to specify input and output delays relative to a clock. An example where virtual clock is applicable is shown in Figure 7-33. The design under analysis gets its clock from *CLK_CORE*, but the clock driving input port *ROW_IN* is *CLK_SAD*. How does one specify the IO constraint on input port *ROW_IN* in such cases? The same issue occurs on the output port *STATE_O*.

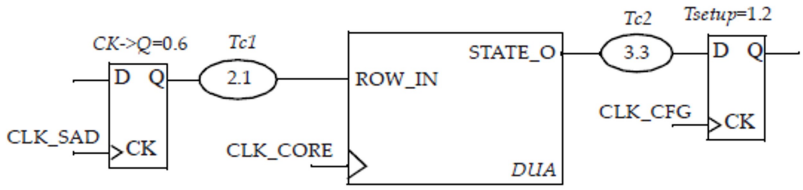


Figure 7-33 *Virtual clocks for CLK_SAD and CLK_CFG.*

To handle such cases, a virtual clock can be defined with no specification of the source port or pin. In the example of Figure 7-33, the virtual clock is defined for *CLK_SAD* and *CLK_CFG*.

```
create_clock -name VIRTUAL_CLK_SAD -period 10 -waveform {2 8}


create_clock -name VIRTUAL_CLK_CFG -period 8 \
  -waveform {0 4}
create_clock -period 10 [get_ports CLK_CORE]
```

Having defined these virtual clocks, the IO constraints can be specified relative to this virtual clock.

```
set_input_delay -clock VIRTUAL_CLK_SAD -max 2.7 \
  [get_ports ROW_IN]

set_output_delay -clock VIRTUAL_CLK_CFG -max 4.5 \
  [get_ports STATE_O]
```

Figure 7-34 shows the timing relationships on the input path. This constrains the input path in the design under analysis to be 5.3ns or less.

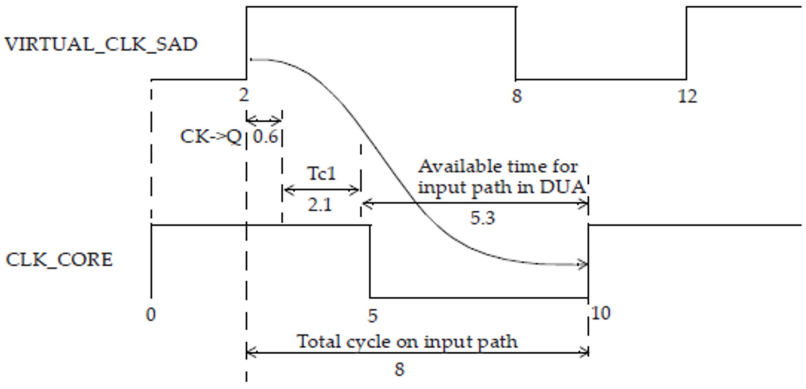


Figure 7-34 *Virtual clock and core clock waveform for input path.*

Refining the timing path
set_case_analysis
set_disable_timing
set_false_path
set_multicycle_path

Refining the Timing Analysis

Four common commands that are used to constrain the analysis space are:

i. set_case_analysis: Specifies constant value on a pin of a cell, or on an input port.

ii. set_disable_timing: Breaks a timing arc of a cell.

iii. set_false_path: Specifies paths that are not real which implies that these paths are not checked in STA.

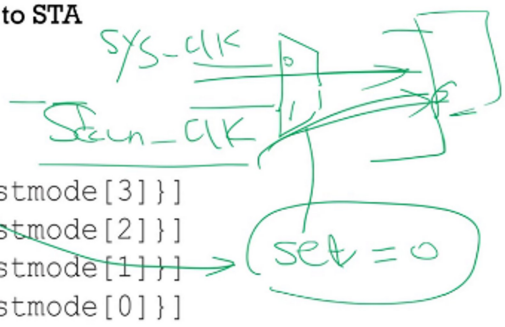
iv. set_multicycle_path: Specifies paths that can take longer than one clock cycle

set_case_analysis
在某些特定条件下需要把scan_clk置为0，是为了缩小STA遍历的变量

Refining the Timing Analysis

In a design, certain signals have a constant value in a specific mode of the chip.

For example, if a chip has DFT logic in it, then the TEST pin of the chip should be at 0 in normal functional mode. It is often useful to specify such constant values to STA

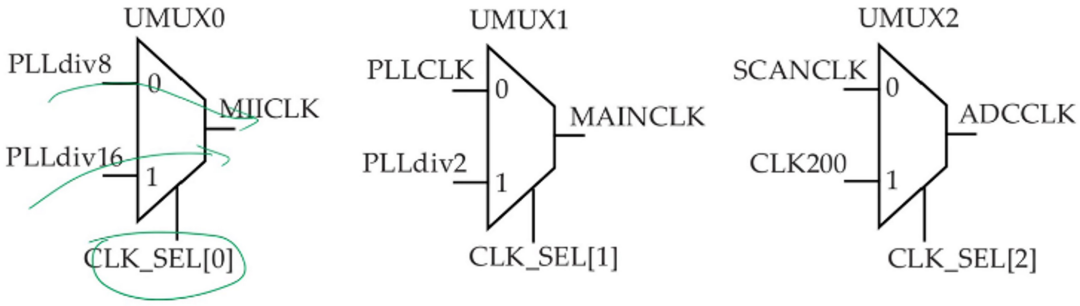



```
set_case_analysis 0 TEST

set_case_analysis 0 [get_ports {testmode[3]}]
set_case_analysis 0 [get_ports {testmode[2]}]
set_case_analysis 0 [get_ports {testmode[1]}]
set_case_analysis 0 [get_ports {testmode[0]}]
```

Refining the Timing Analysis

Another common application of case analysis is when the design can run on multiple clocks, and the selection of the appropriate clock is controlled by multiplexers. To make STA analysis easier and reduce CPU run time, it is beneficial to do STA for each clock selection separately.



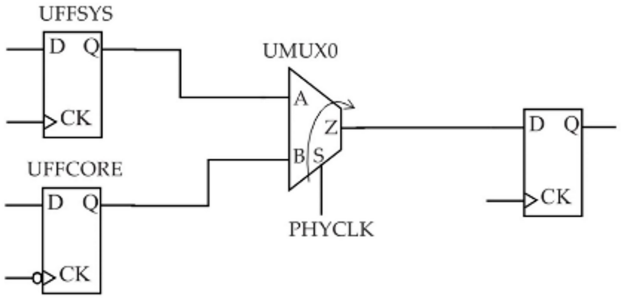

```
set_case_analysis 1 UCORE/UMUX0/CLK_SEL[0]
set_case_analysis 1 UCORE/UMUX1/CLK_SEL[1]
set_case_analysis 0 UCORE/UMUX2/CLK_SEL[2]
```

S to Z没有timing arc，那么需要加入disable_timing来指定没有timing arc的情况

Refining the Timing Analysis

In some situations, it is possible that a certain path through a cell cannot occur.

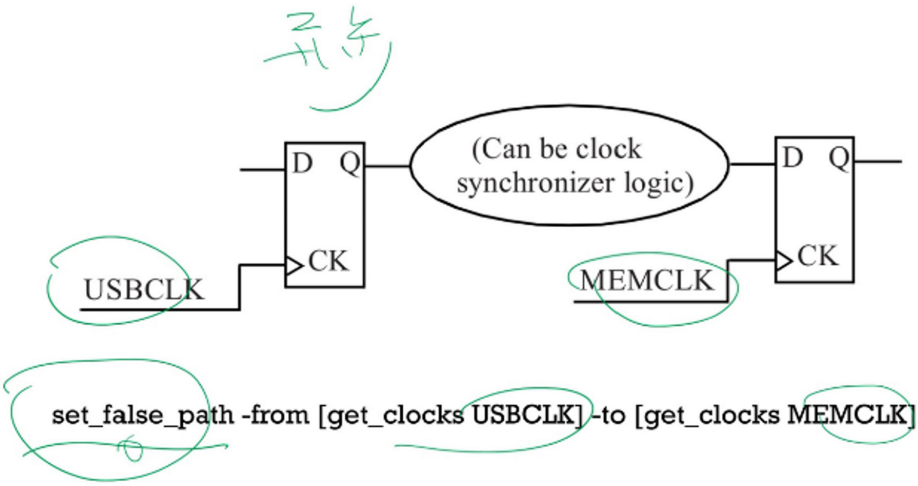
Such a timing arc can be broken by using the set_disable_timing SDC command.




```
set_disable_timing -from S -to Z [get_cells UMUX0]
```

False path
异步时钟

Refining the Timing Analysis

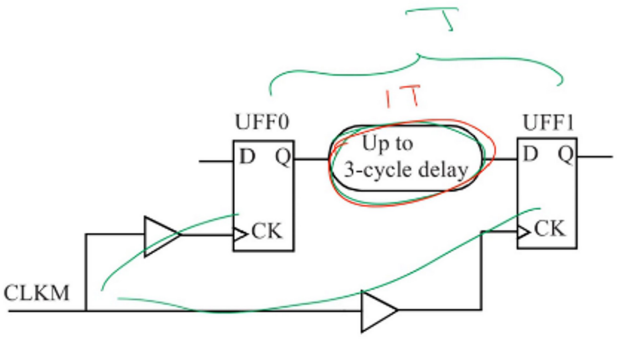


多周期路径

这里用了3个周期检查

RTL designer会告诉backend designer哪个timing path是多周期，哪个是false path

Refining the Timing Analysis




```
create_clock -name CLKM -period 10 [get_ports CLKM]
set_multicycle_path 3 -setup -from [get_pins UFF0/Q] -to [get_pins UFF1/D]
```

掌握SDC writing && correction