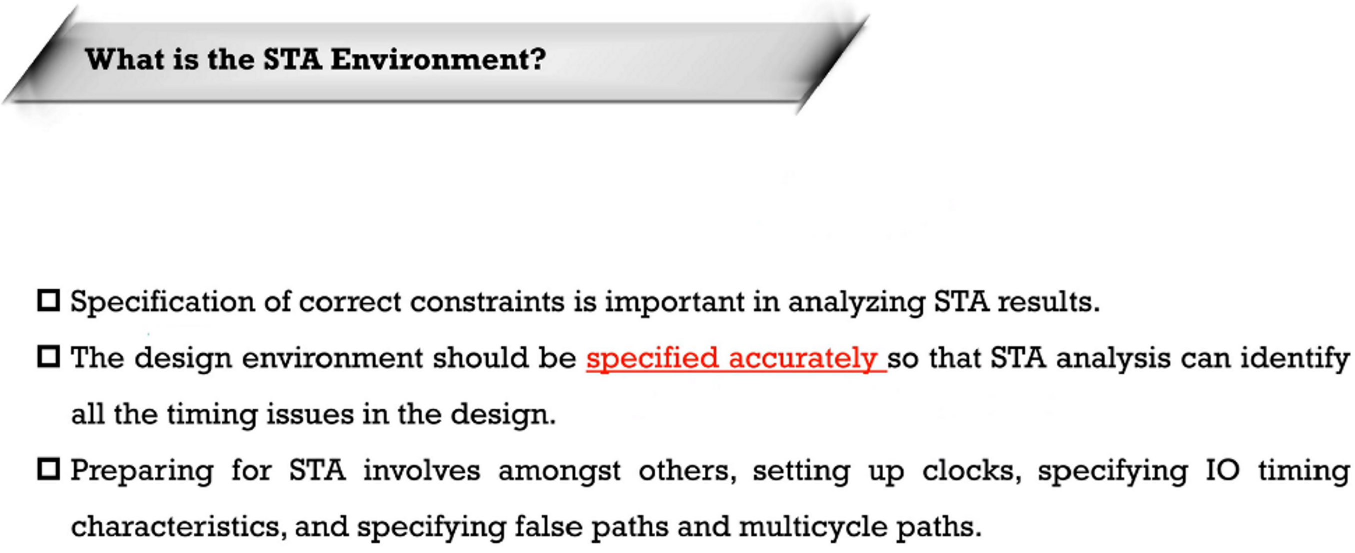
SDC constraints I

2022年6月9日 14:03

STA environment - SDC constraints

建立正确的约束条件，才能check setup和hold是否满足要求



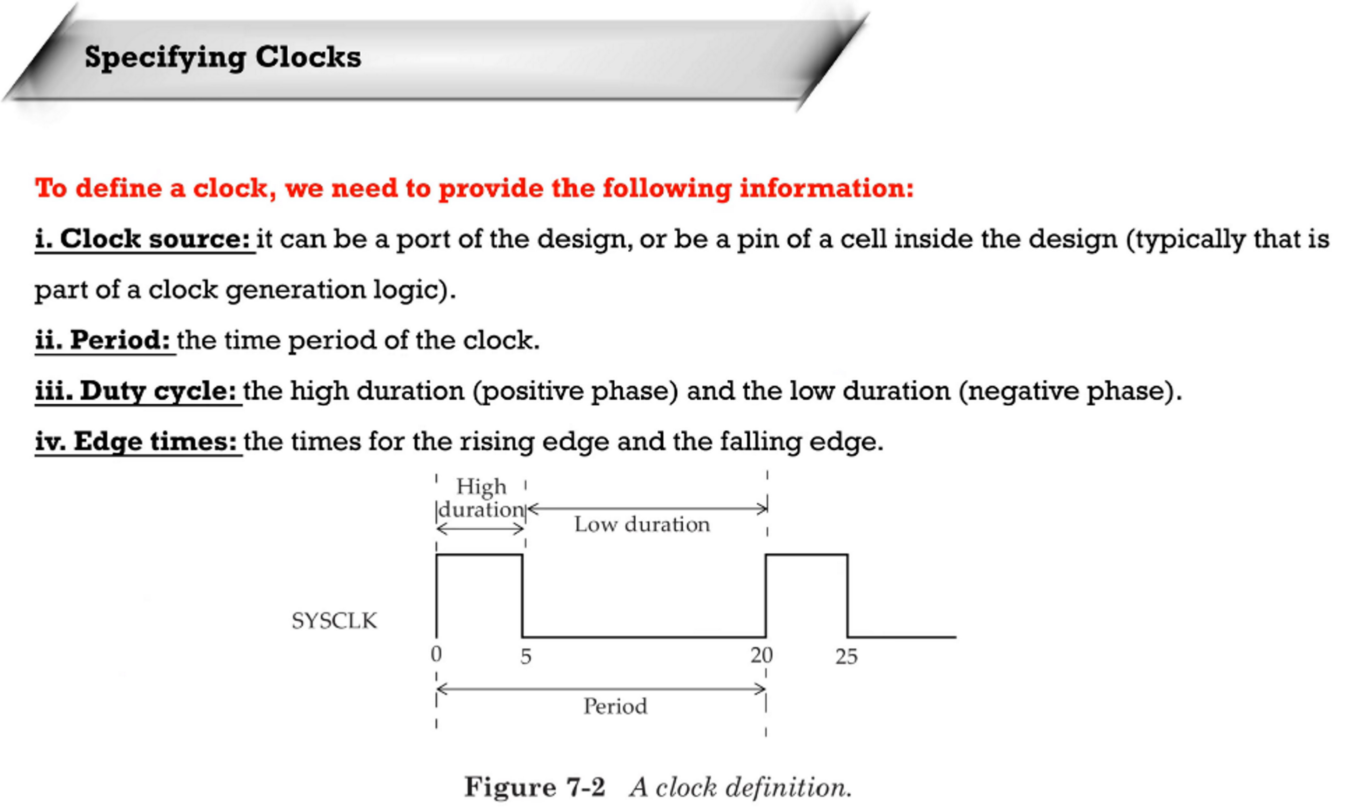
What is the STA Environment?

□ Specification of correct constraints is important in analyzing STA results.

□ The design environment should be specified accurately so that STA analysis can identify all the timing issues in the design.

□ Preparing for STA involves amongst others, setting up clocks, specifying IO timing characteristics, and specifying false paths and multicycle paths.

定义时钟



Specifying Clocks

To define a clock, we need to provide the following information:

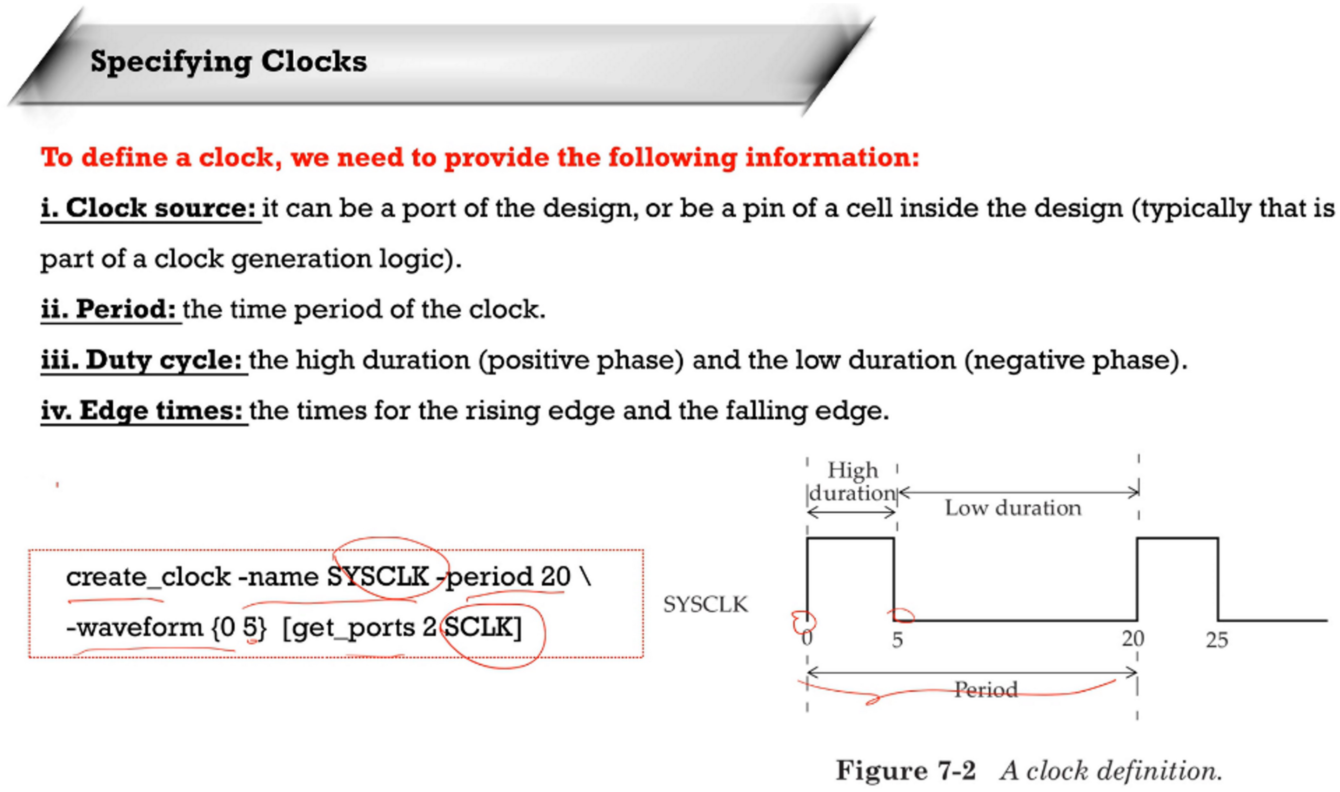
i. Clock source: it can be a port of the design, or be a pin of a cell inside the design (typically that is part of a clock generation logic).

ii. Period: the time period of the clock.

iii. Duty cycle: the high duration (positive phase) and the low duration (negative phase).

iv. Edge times: the times for the rising edge and the falling edge.

Figure 7-2 *A clock definition.*



Specifying Clocks

To define a clock, we need to provide the following information:

i. Clock source: it can be a port of the design, or be a pin of a cell inside the design (typically that is part of a clock generation logic).

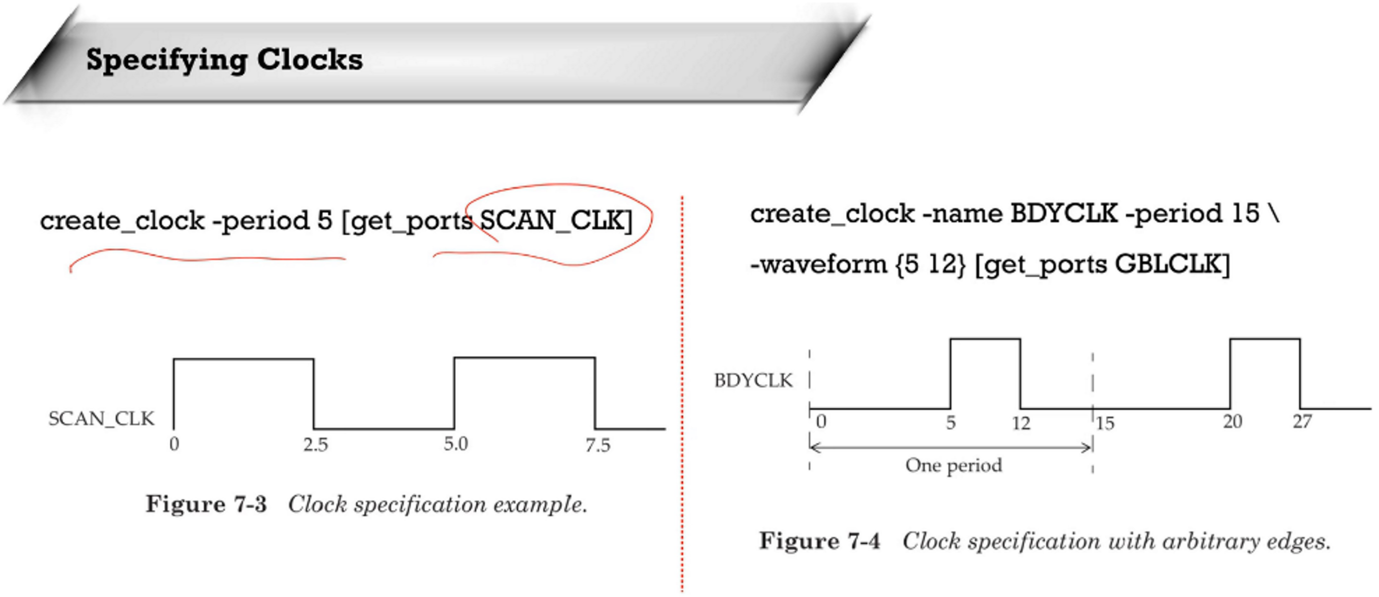
ii. Period: the time period of the clock.

iii. Duty cycle: the high duration (positive phase) and the low duration (negative phase).

iv. Edge times: the times for the rising edge and the falling edge.

```
create_clock -name SYSCLK -period 20 \
-waveform {0 5}  [get_ports 2 SCLK]
```

Figure 7-2 *A clock definition.*



Specifying Clocks

```
create_clock -period 5 [get_ports SCAN_CLK]
```

Figure 7-3 *Clock specification example.*

```
create_clock -name BDYCLK -period 15 \
-waveform {5 12} [get_ports GBLCLK]
```

Figure 7-4 *Clock specification with arbitrary edges.*

Define clock uncertainty to reduce the effective clock period and create more pessimism

Specifying Clocks-Clock Uncertainty

- The timing uncertainty of a clock period can be specified using the set_clock_uncertainty specification.
- The uncertainty can be used to model various factors that can ***reduce*** the effective clock period.
- These factors can be *the clock jitter and any other pessimism* that one may want to include for timing analysis.

Specifying Clocks-Clock Uncertainty

```
set_clock_uncertainty -setup 0.2 [get_clocks CLK_CONFIG]
set_clock_uncertainty -hold 0.05 [get_clocks CLK_CONFIG]
```

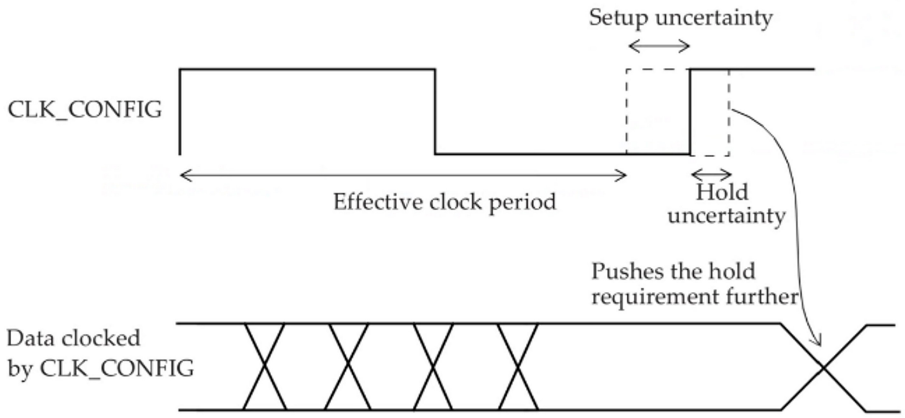
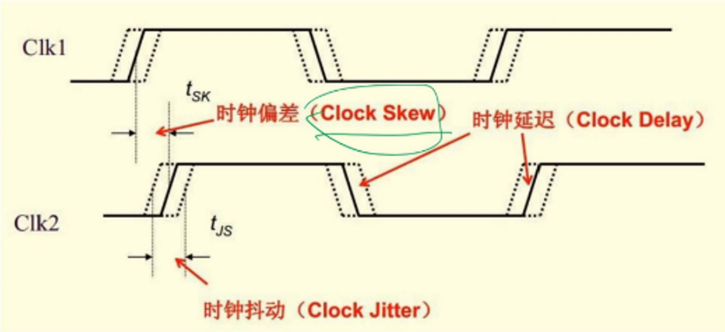


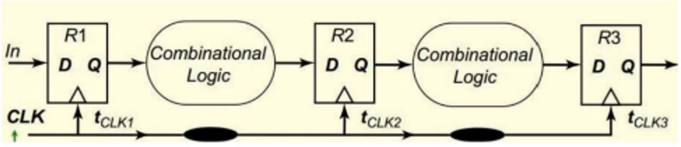
Figure 7-7 *Specifying clock uncertainty.*

Must reduce clock skew (CLK2 - CLK1)
clock jitter for single clock edge

Specifying Clocks-Clock Uncertainty



≥0 ≤0



Clock - source latency && network latency

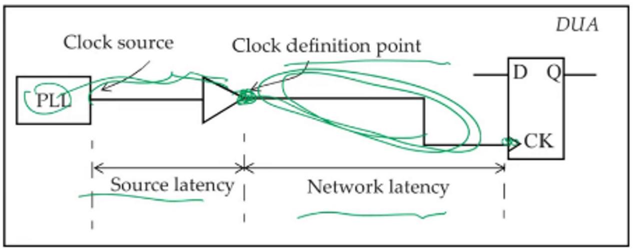
Specifying Clocks-Clock Latency

There are two types of clock latencies: **network latency** and **source latency**.

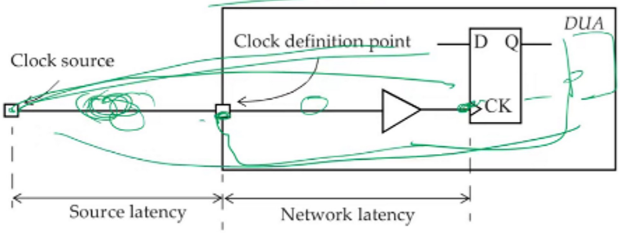
- □ **Network latency** is the delay from the clock definition point (create_clock) to the clock pin of a flip-flop.
- □ **Source latency**, also called insertion delay, is the delay from the clock source to the clock definition point.

Source latency could represent either on-chip or off-chip latency. The total clock latency at the clock pin of a flip-flop is the sum of the source and network latencies.

CDPR



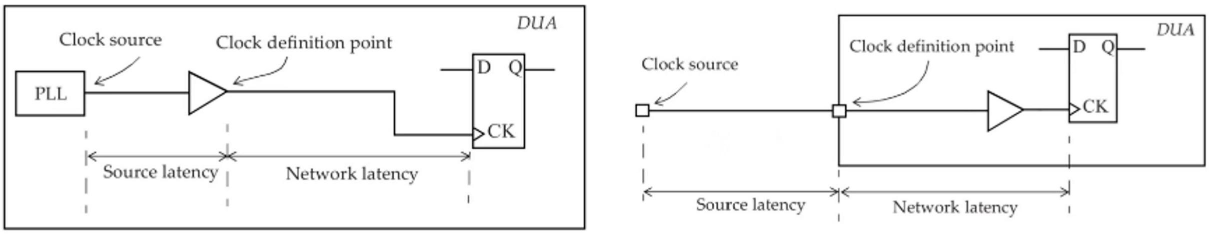
(a) On-chip clock source.



(b) Off-chip clock source.

Specifying Clocks-Clock Latency

```
# Specify a network latency (no -source option) of 0.8ns for  rise, fall, max and min:
set_clock_latency 0.8 [get_clocks CLK_CONFIG]
# Specify a source latency:
set_clock_latency 1.9 -source [get_clocks SYS_CLK]
# Specify a min source latency:
set_clock_latency 0.851 -source -min [get_clocks CFG_CLK]
# Specify a max source latency:
set_clock_latency 1.322 -source -max [get_clocks CFG_CLK]
```



(a) On-chip clock source.

(b) Off-chip clock source.

根据时钟树的建立节点分析
时钟树建好了，network delay非常精确

Specifying Clocks-Clock Latency

- □ One important distinction to observe between source and network latency is that once a clock tree is built for a design, the network latency can be ignored (assuming set_propagated_clock command is specified).
- □ However, the source latency remains even after the clock tree is built.
- □ The network latency is an **estimate** of the delay of the clock tree **prior** to clock tree synthesis.
- □ After clock tree synthesis, the total clock latency from clock source to a clock pin of a flip-flop is the source latency plus the actual delay of the clock tree from the clock definition point to the flip-flop.

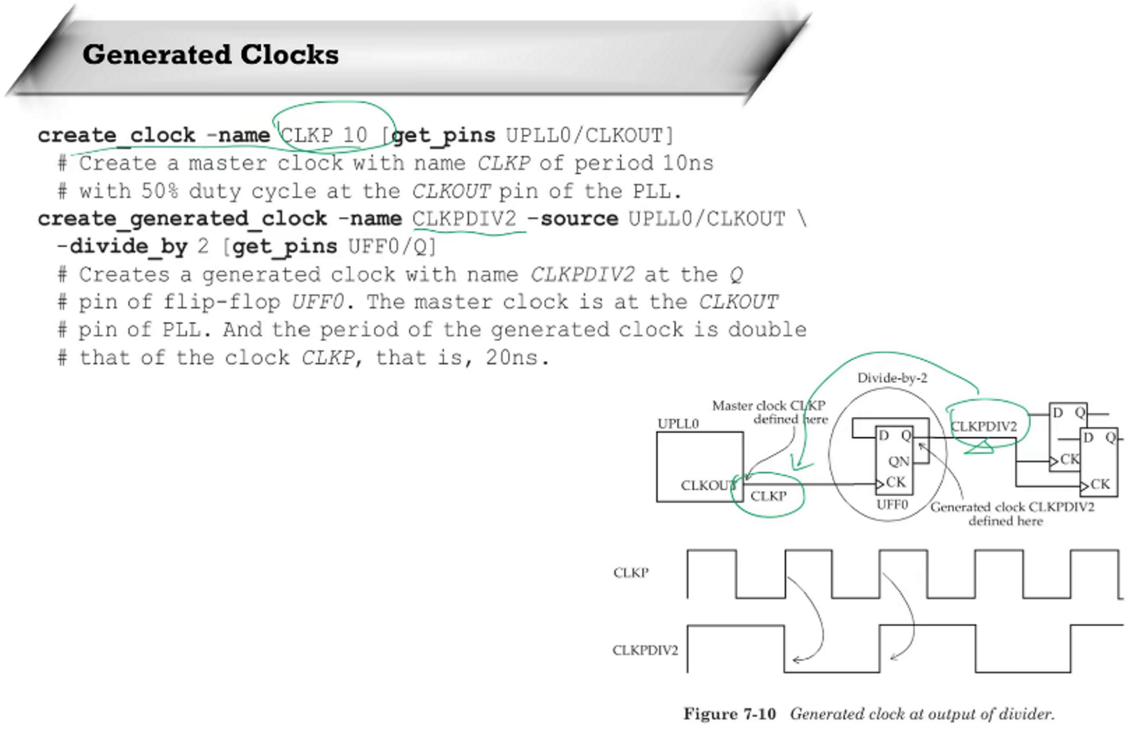
生成时钟

Generated Clocks

- □ A generated clock is a clock derived from a master clock. A master clock is a clock defined using the create_clock specification.
- □ When a new clock is generated in a design that is based on a master clock, the new clock can be defined as a generated clock.

This definition is needed as STA does not know that the clock period has changed at the output of the divide-by logic, and more importantly what the new clock period is.

generated clocks是master clocks的二分频



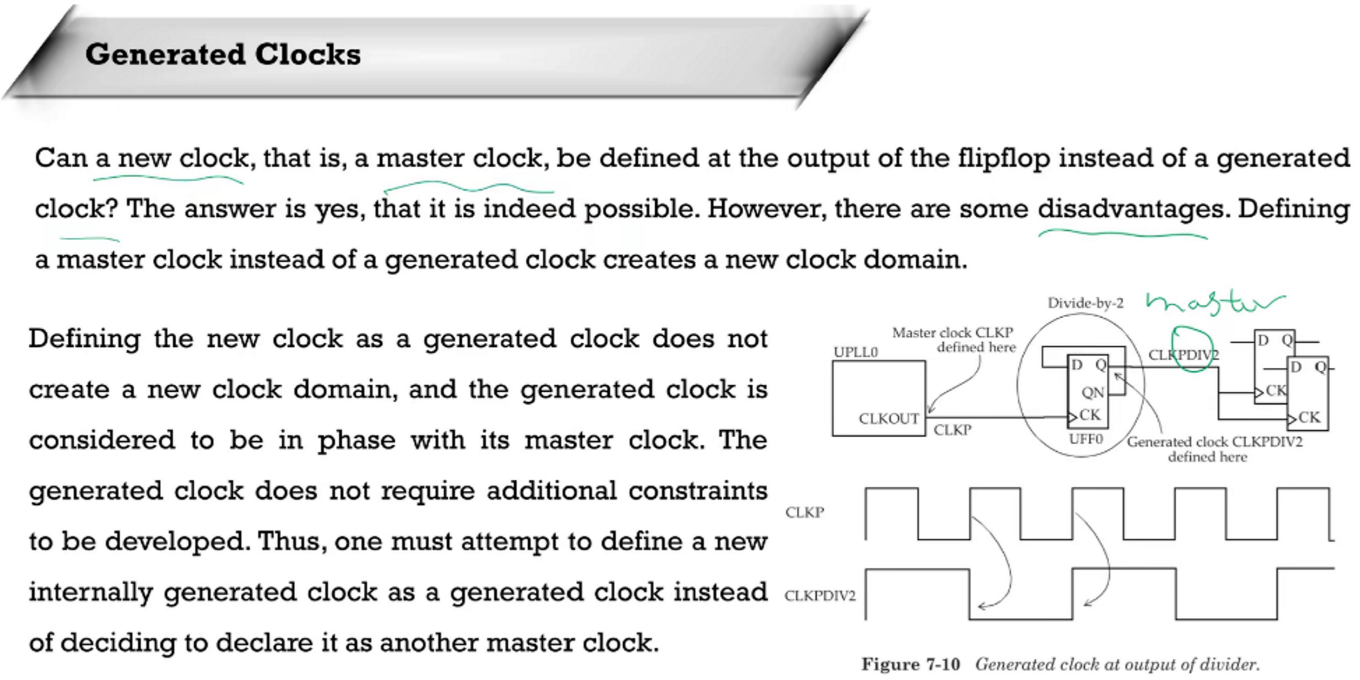
Generated Clocks

```
create_clock -name CLKP 10 [get_pins UPLL0/CLKOUT]
  # Create a master clock with name CLKP of period 10ns
  # with 50% duty cycle at the CLKOUT pin of the PLL.
create_generated_clock -name CLKPDIV2 -source UPLL0/CLKOUT \
  -divide_by 2 [get_pins UFF0/Q]
  # Creates a generated clock with name CLKPDIV2 at the Q
  # pin of flip-flop UFF0. The master clock is at the CLKOUT
  # pin of PLL. And the period of the generated clock is double
  # that of the clock CLKP, that is, 20ns.
```

Divide-by-2
Master clock CLKP defined here
UPLL0
CLKOUT
CLKP
D Q
QN
CK
UFF0
CLKPDIV2
Generated clock CLKPDIV2 defined here
D Q
CK
D Q
CK
CLKP
CLKPDIV2

Figure 7-10 *Generated clock at output of divider.*

生成时钟，如果被定义成新的master clock会失去同步时钟的继承性，而被认为是异步时钟



Generated Clocks

Can a new clock, that is, a master clock, be defined at the output of the flipflop instead of a generated clock? The answer is yes, that it is indeed possible. However, there are some disadvantages. Defining a master clock instead of a generated clock creates a new clock domain.

Defining the new clock as a generated clock does not create a new clock domain, and the generated clock is considered to be in phase with its master clock. The generated clock does not require additional constraints to be developed. Thus, one must attempt to define a new internally generated clock as a generated clock instead of deciding to declare it as another master clock.

Figure 7-10 *Generated clock at output of divider.*

Generated Clocks

Another important difference between a master clock and a generated clock is the notion of clock origin. In a master clock, the origin of the clock is at the point of definition of the master clock. In a generated clock, the clock origin is that of the master clock and not that of the generated clock.

This implies that in a clock path report, the start point of a clock path is always the master clock definition point. This is a big advantage of a generated clock over defining a new master clock as the source latency is not automatically included for the case of a new master clock.

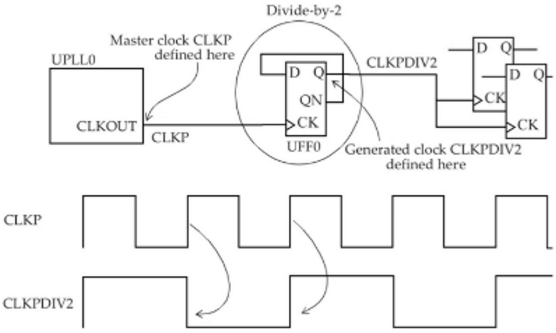
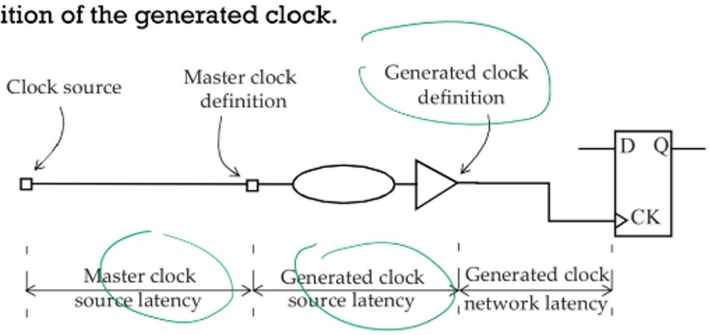


Figure 7-10 *Generated clock at output of divider.*

Generated Clocks

Clock latencies can be specified for generated clocks as well.

A source latency specified on a generated clock specifies the latency from the definition of the master clock to the definition of the generated clock.

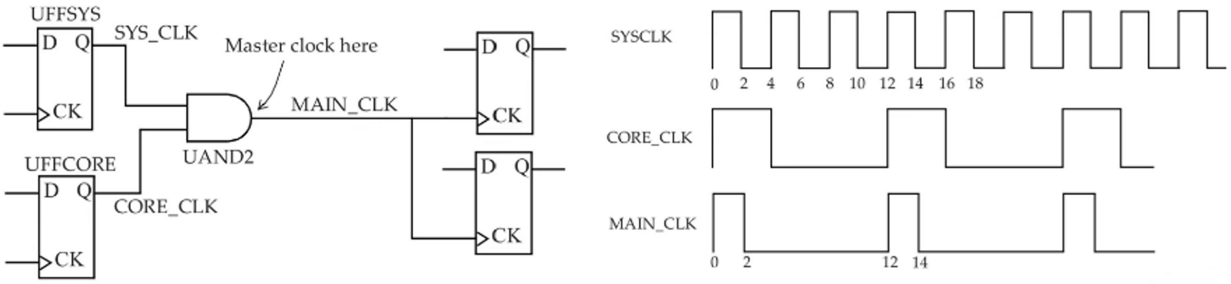


The total clock latency to a clock pin of a flop-flop being driven by a generated clock is thus the sum of the source latency of the master clock, the source latency of the generated clock and the network latency of the generated clock.

不是所有情况都定义为生成时钟

Generated Clocks

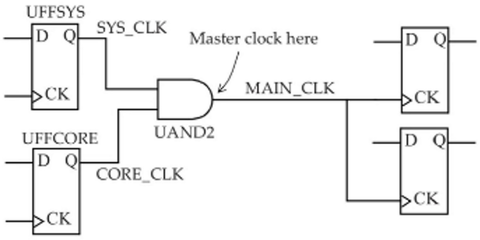
If the input to the and cell are both clocks, then it is safe to define a new main clock at the output of the and cell, since it is highly unlikely that the output of the cell has any phase relationship with either of the input clocks.



Generated Clocks

If the input to the and cell are both clocks, then it is safe to define a new main clock at the output of the and cell, since it is highly unlikely that the output of the cell has any phase relationship with either of the input clocks.

```
create_clock -name SYS_CLK -period 4 -waveform {0 2} \
  [get_pins UFFSYS/Q]
create_clock -name CORE_CLK -period 12 -waveform {0 4} \
  [get_pins UFFCORE/Q]
create_clock -name MAIN_CLK -period 12 -waveform {0 2} \
  [get_pins UAND2/Z]
  # Create a master clock instead of a generated clock
  # at the output of the and cell.
```



低速时钟通过锁相环生成高速时钟，再分频成生成时钟

Generated Clocks

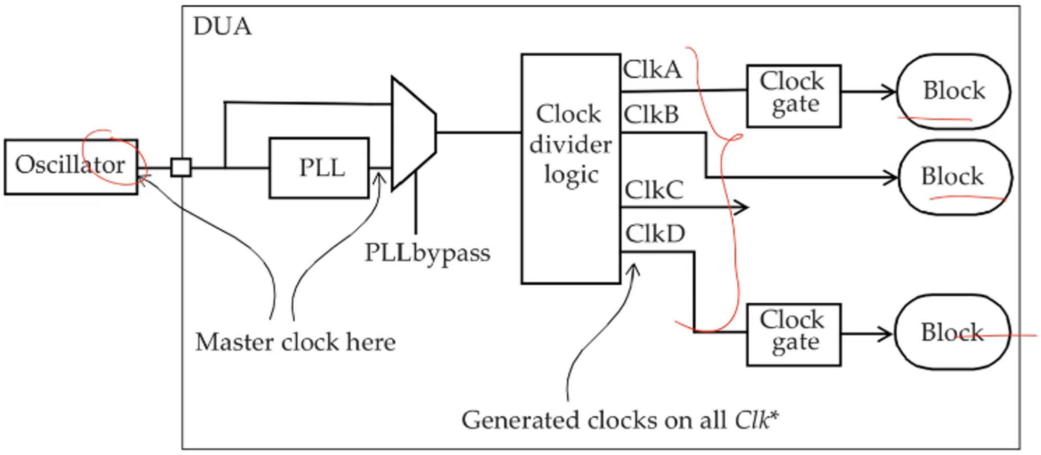


Figure 7-20 *Clock distribution in a typical ASIC.*