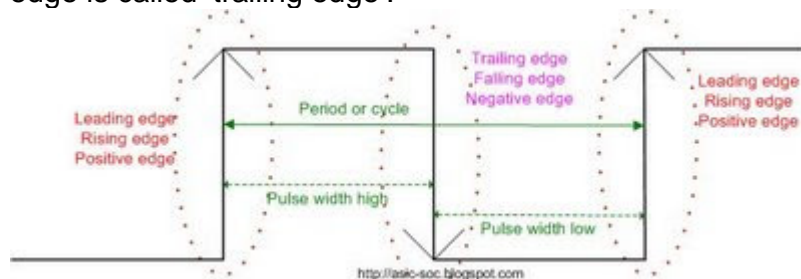


CLOCK DEFINITIONS

Rising and falling edge of the clock

For a +ve edge triggered design +ve (or rising) edge is called 'leading edge' whereas -ve (or falling) edge is called 'trailing edge'.

For a -ve edge triggered design -ve (or falling) edge is called 'leading edge' whereas +ve (or rising) edge is called 'trailing edge'.



basic clock

Minimum pulse width of the clock can be checked in PrimeTime by using commands given below:

```
set_min_pulse_width -high 2.5 [all_clocks]
```

```
set_min_pulse_width -low 2.0 [all_clocks]
```

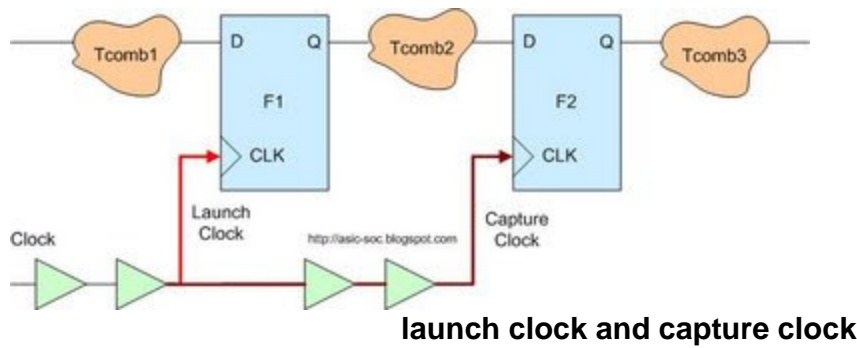
These checks are generally carried out for post layout timing analysis. Once these commands are set, PrimeTime checks for high and low pulse widths and reports any violations.

Capture Clock Edge

The edge of the clock for which data is detected is known as capture edge.

Launch Clock Edge

This is the edge of the clock wherein data is launched in previous flip flop and will be captured at this flip flop.



Skew

Skew is the difference in arrival of clock at two consecutive pins of a sequential element is called skew. Clock skew is the variation at arrival time of clock at destination points in the clock network. The difference in the arrival of clock signal at the clock pin of different flops.

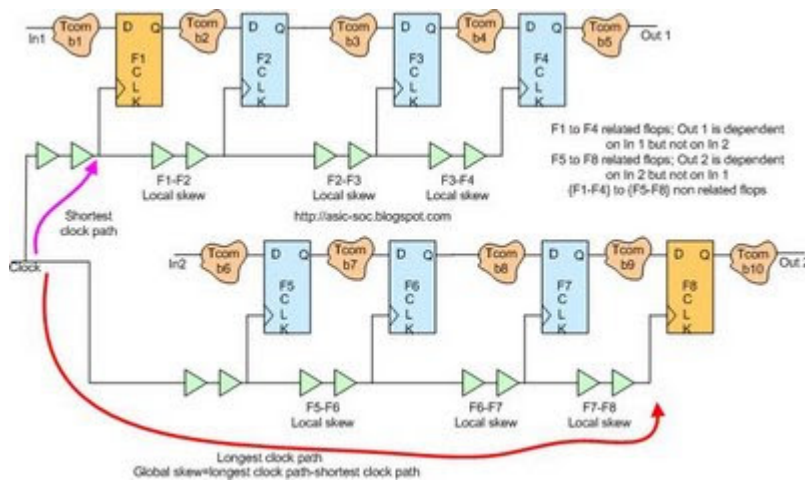
Two types of skews are defined: Local skew and Global skew.

Local skew

Local skew is the difference in the arrival of clock signal at the clock pin of related flops.

Global skew

Global skew is the difference in the arrival of clock signal at the clock pin of non related flops. This also defined as the difference between shortest clock path delay and longest clock path delay reaching two sequential elements.



Skew can be positive or negative. When data and clock are routed in same direction then it is **Positive skew**. When data and clock are routed in opposite direction then it is **negative skew**.

Positive Skew

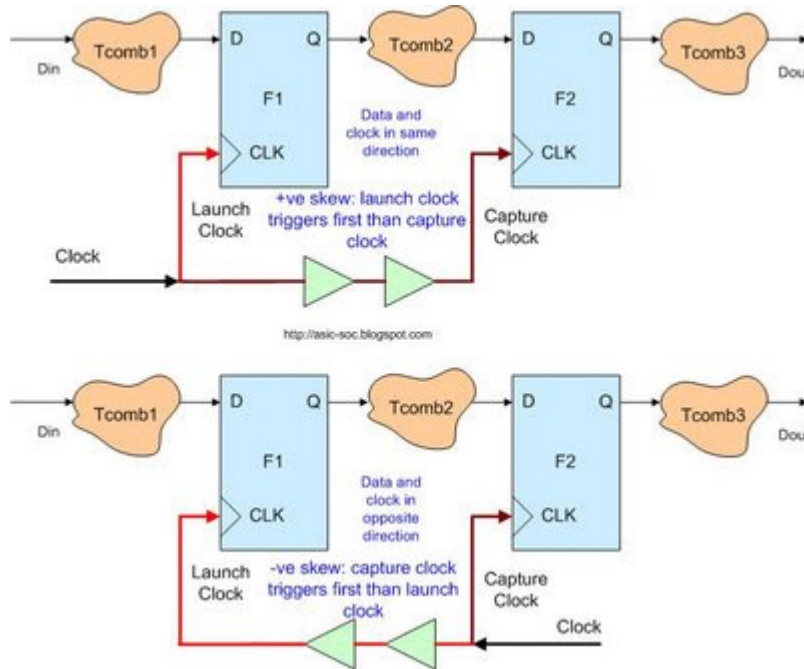
If capture clock comes late than launch clock then it is called +ve skew.

Clock and data both travel in same direction.

When data and clock are routed in same direction then it is **Positive skew**.

+ve skew can lead to hold violation.

+ve skew improves setup time.



positive skew negative skew

Negative Skew

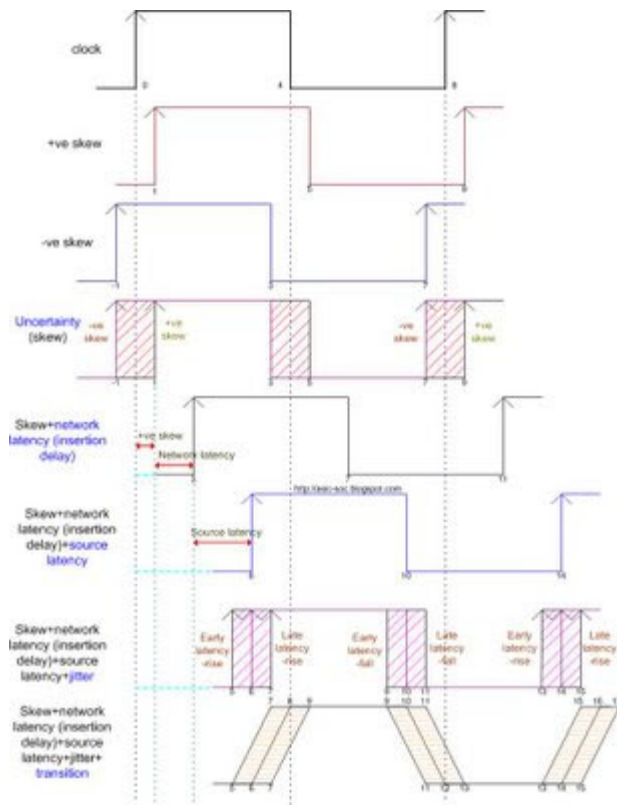
If capture clock comes early than launch clock it is called -ve skew. Clock and data travel in opposite direction. When data and clock are routed in opposite then it is **negative skew**. -ve skew can lead to setup violation. -ve skew improves hold time. (Effects of skew on setup and hold will be discussed in detail in forthcoming articles)

Uncertainty

Clock uncertainty is the time difference between the arrivals of clock signals at registers in one clock domain or between domains.

Pre-layout and Post-layout Uncertainty

Pre CTS uncertainty is clock skew, clock Jitter and margin. After CTS skew is calculated from the actual propagated value of the clock. We can have some margin of skew + Jitter.



timing diagram depicting skew, latency, jitter

Clock latency

Latency is the delay of the clock source and clock network delay.

Clock source delay is the time taken to propagate from ideal waveform origin point to clock definition point. Clock network latency is the delay from clock definition point to register clock pin.

Pre CTS Latency and Post CTS Latency

Latency is the summation of the Source latency and the Network latency. Pre CTS estimated latency will be considered during the synthesis and after CTS propagated latency is considered.

Source Delay or Source Latency

It is known as source latency also. It is defined as "the delay from the clock origin point to the clock definition point in the design".

Delay from clock source to beginning of clock tree (i.e. clock definition point).

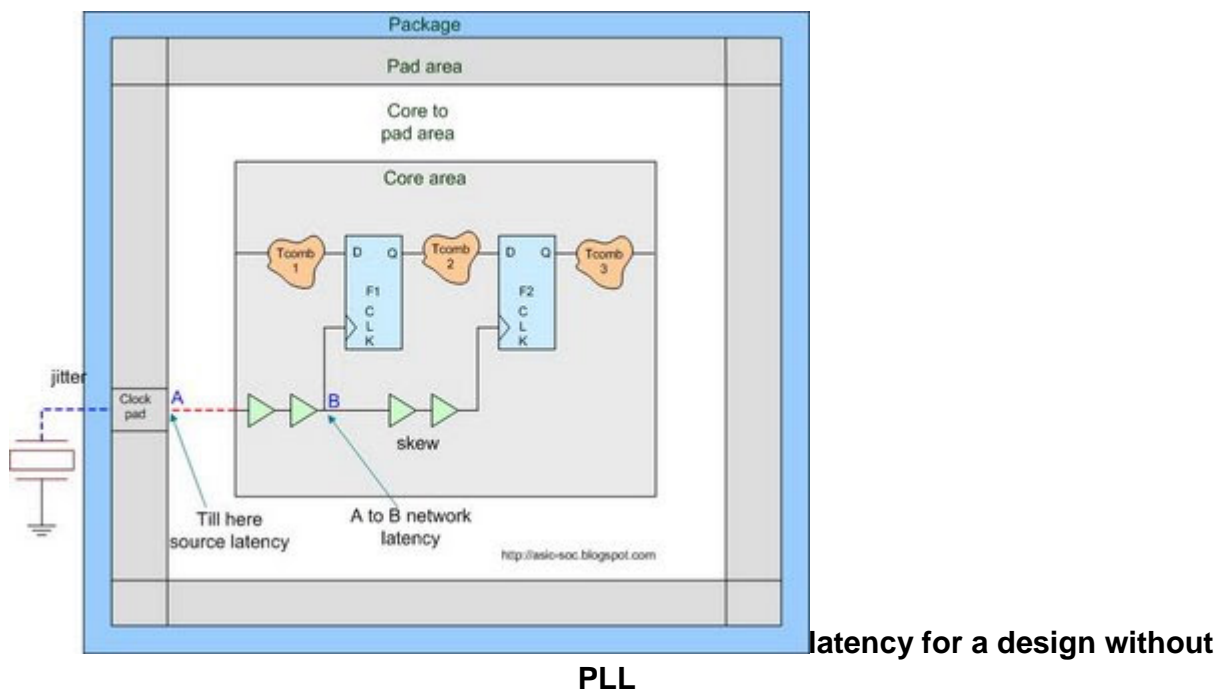
The time a clock signal takes to propagate from its ideal waveform origin point to the clock definition point in the design.

Network Delay (latency) or Insertion Delay

It is also known as Insertion delay or Network latency. It is defined as "the delay from the clock definition point to the clock pin of the register".

The time clock signal (rise or fall) takes to propagate from the clock definition point to a register clock pin.

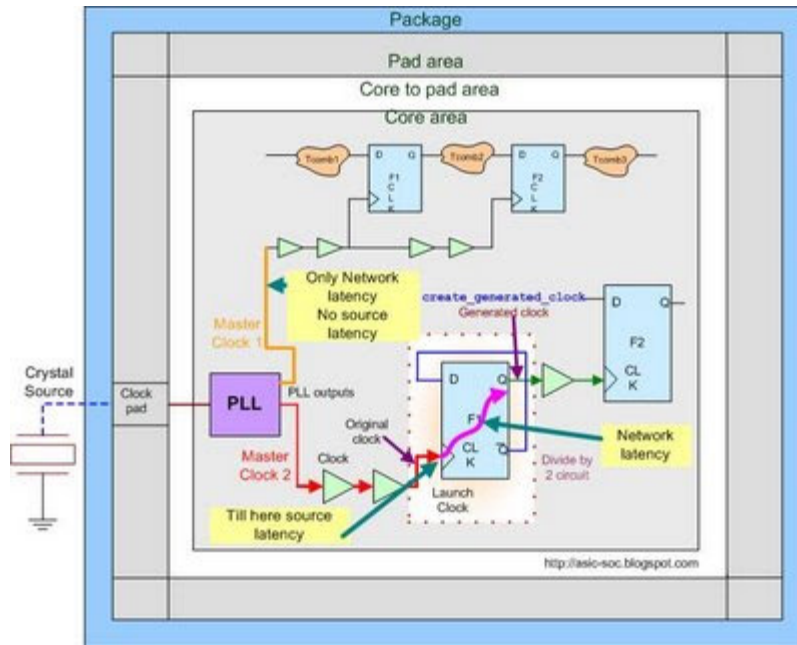
Figure below shows example of latency for a design without PLL.



The latency definitions for designs with PLL are slightly different.

Figure below shows latency specifications of such kind of designs.

Latency from the PLL output to the clock input of generated clock circuitry becomes source latency. From this point onwards till generated clock divides to flops is now known as network latency. Here we can observe that part of the network latency is clock to q delay of the flip flop (of divide by 2 circuit in the given example) is known value.



latency for a design with PLL

Jitter

Jitter is the short-term variations of a signal with respect to its ideal position in time.

Jitter is the variation of the clock period from edge to edge. It can vary +/- jitter value.

From cycle to cycle the period and duty cycle can change slightly due to the clock generation circuitry. Jitter can also be generated from PLL known as PLL jitter. Possible jitter values should be considered for proper PLL design. Jitter can be modeled by adding uncertainty regions around the rising and falling edges of the clock waveform.

Sources of Jitter Common sources of jitter include:

- Internal circuitry of the phase-locked loop (PLL)
- Random thermal noise from a crystal
- Other resonating devices
- Random mechanical noise from crystal vibration
- Signal transmitters
- Traces and cables
- Connectors

- Receivers
- [Click here to read more about jitter from Altera.](#)
- [Click here to read what wiki says about jitter.](#)

Multiple Clocks

If more than one clock is used in a design, then they can be defined to have different waveforms and frequencies. These clocks are known as multiple clocks. The logics triggered by each individual clock are then known as “clock domain”.

If clocks have different frequencies there must be a base period over which all waveforms repeat.

Base period is the least common multiple (LCM) of all clock periods

Asynchronous Clocks

In multiple clock domains, if these clocks do not have a common base period then they are called as asynchronous clocks. Clocks generated from two different crystals, PLLs are asynchronous clocks. Different clocks having different frequencies generated from single crystal or PLL are not asynchronous clocks but they are synchronous clocks.

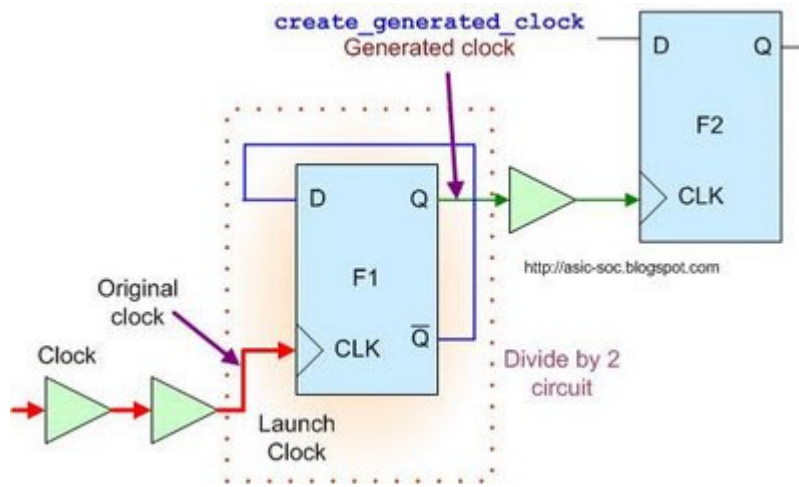
Gated clocks

Clock signals that are passed through some gate other than buffer and inverters are called gated clocks. These clock signals will be under the control of gated logic. Clock gating is used to turn off clock to some sections of design to save power. [Click here to read more about clock gating.](#)

Generated clocks

Generated clocks are the clocks that are generated from other clocks by a circuit within the design such as divider/multiplier circuit.

Static timing analysis tools such as PrimeTime will automatically calculate the latency (delay) from the source clock to the generated clock if the source clock is propagated and you have not set source latency on the generated clock.



generated clock

'Clock' is the master clock and new clock is generated from F1/Q output. Master clock is defined with the constraint 'create_clok'. Unless and until new generated clock is defined as 'generated clock' timing analysis tools won't consider it as generated clock. Hence to accomplish this requirement use "create_generated_clock" command. 'CLK' pin of F1 is now treated as clock definition point for the new generated clock. Hence clock path delay till F1/CLK contributes source latency whereas delay from F1/CLK contributes network latency.

Virtual Clocks

Virtual clock is the clock which is logically not connected to any port of the design and physically doesn't exist. A virtual clock is used when a block does not contain a port for the clock that an I/O signal is coming from or going to. Virtual clocks are used during optimization; they do not really exist in the circuit.

Virtual clocks are clocks that exist in memory but are not part of a design. Virtual clocks are used as a reference for specifying input and output delays relative to a clock. This means there is no actual clock source in the design. Assume the block to be synthesized is "Block_A". The clock signal, "VCLK", would be a virtual clock. The input delay and output delay would be specified relative to the virtual clock.

Source : <http://asic-soc.blogspot.in/2009/01/clock-definitions.html>