

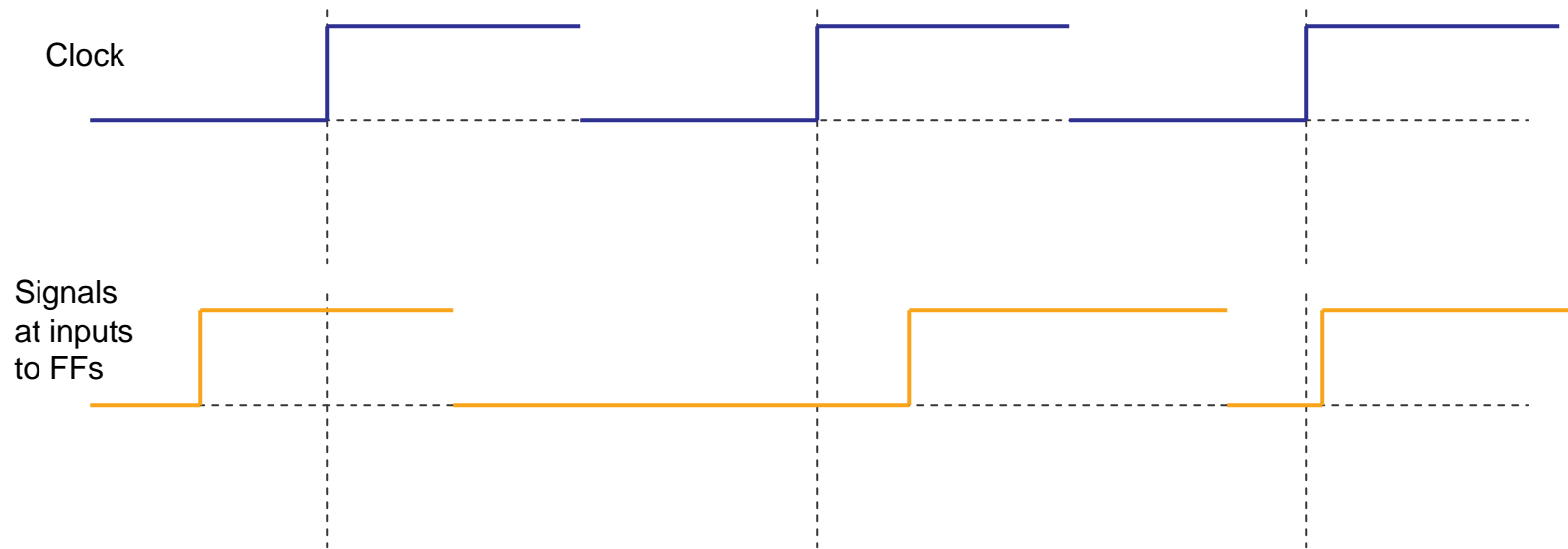
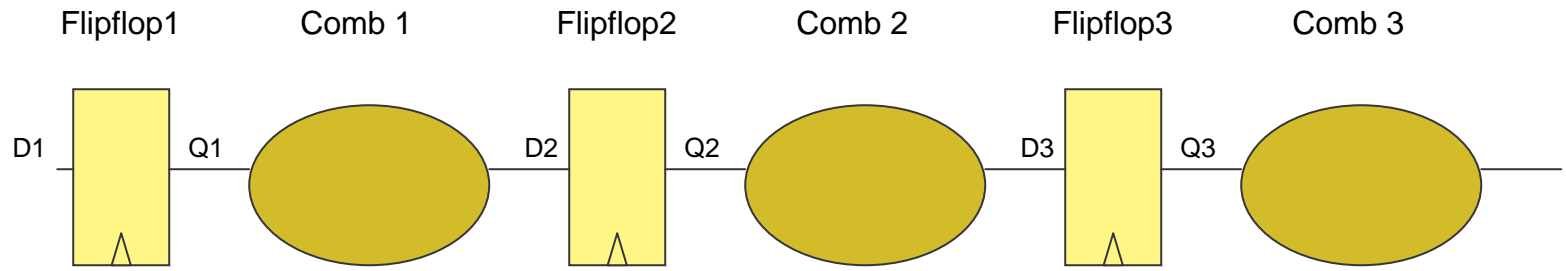
Tutorial:

# Understanding Setup and Hold Violations in Digital System Design

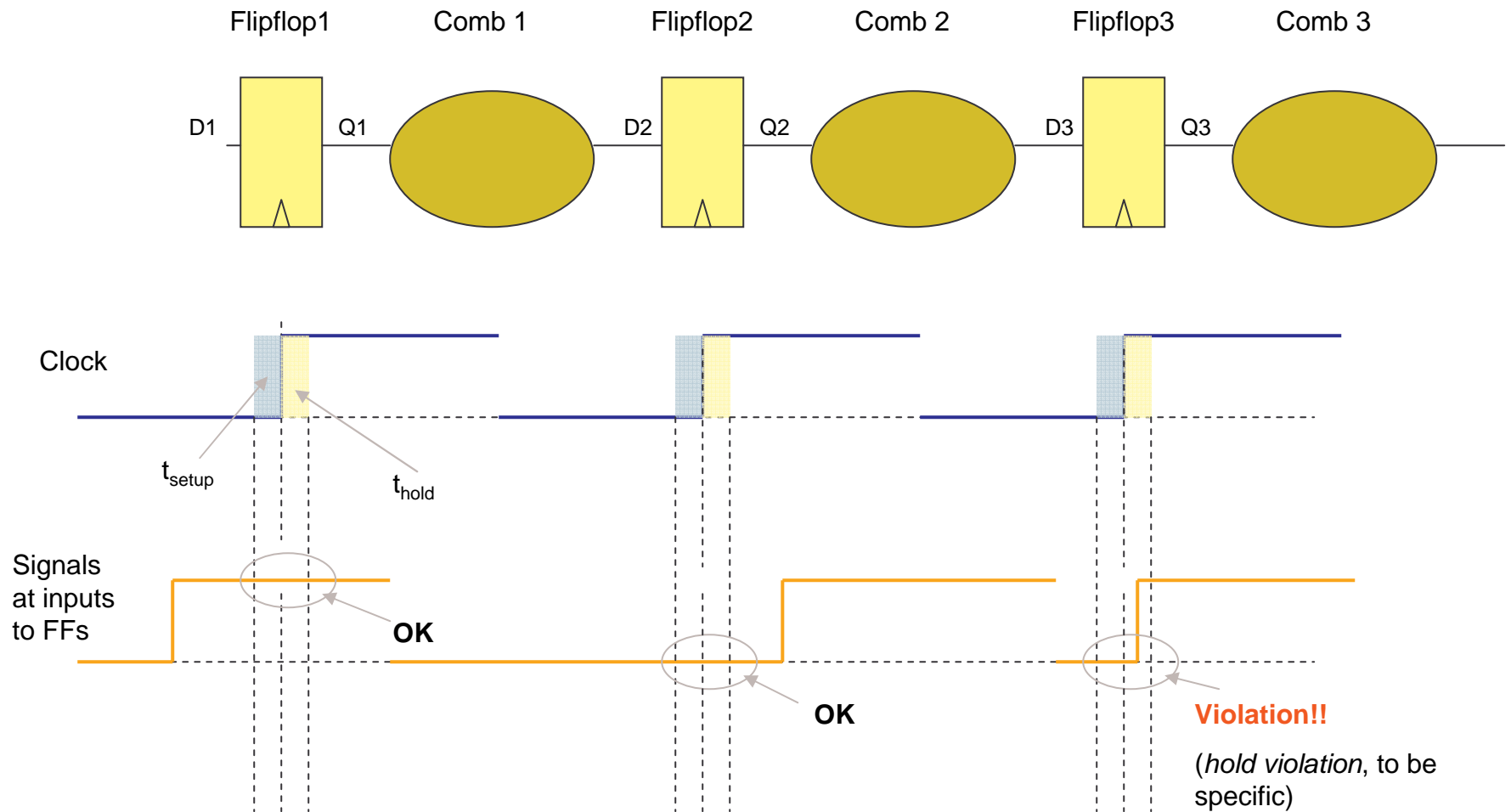
Anil Krishna

Student, NCSU

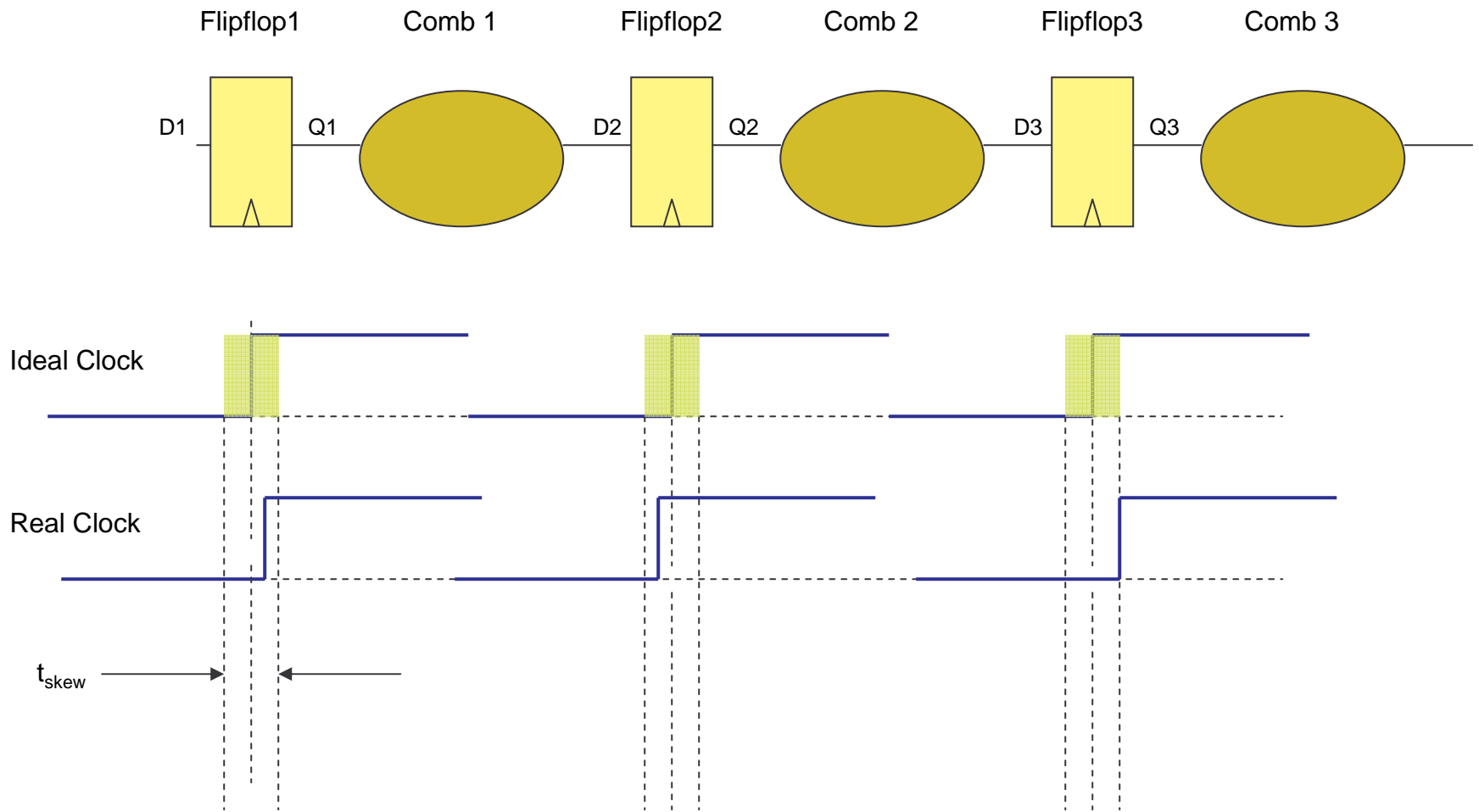
February 2008



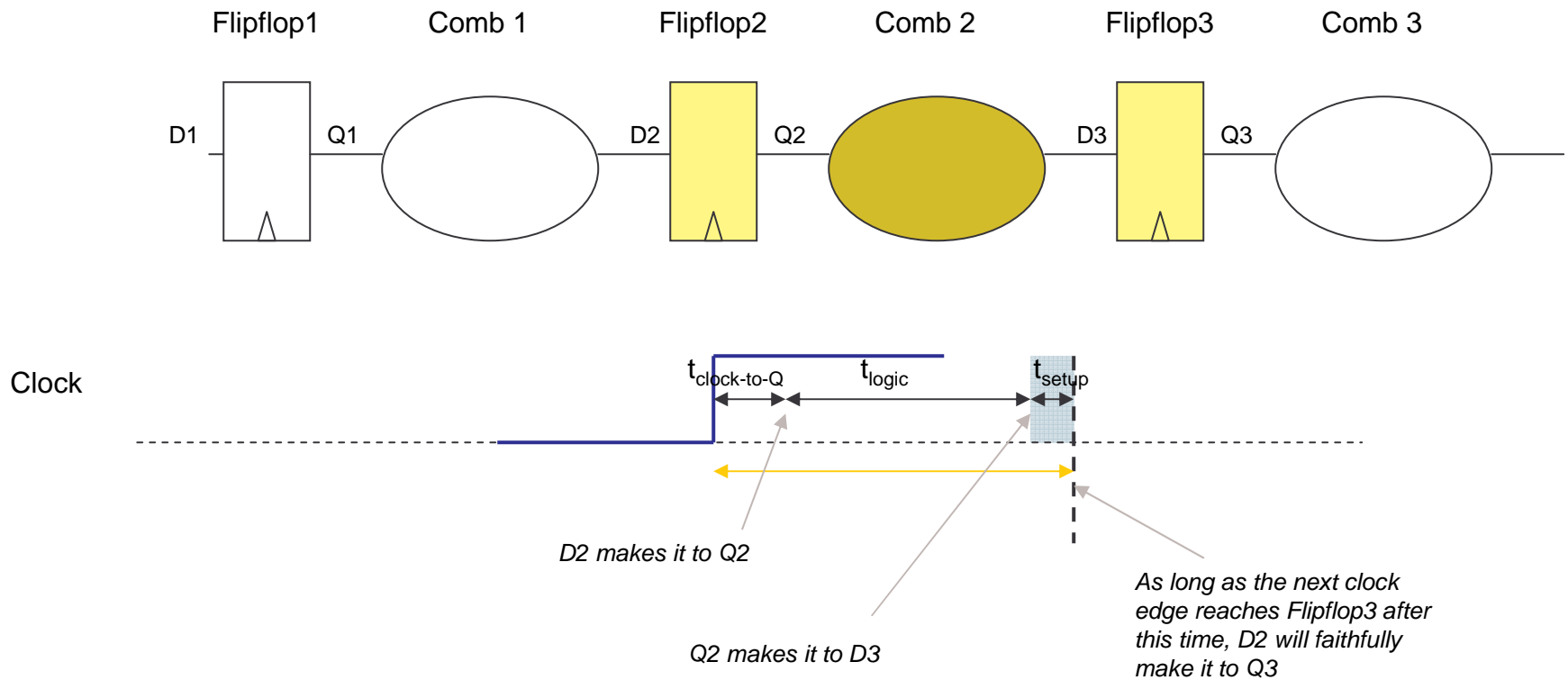
The flipflops shown are positive edge triggered, i.e. on the positive edge of the clock, they takes the value of the signal at its input and send it to the flipflop's output after a small delay called the  $t_{\text{clock-to-Q}}$



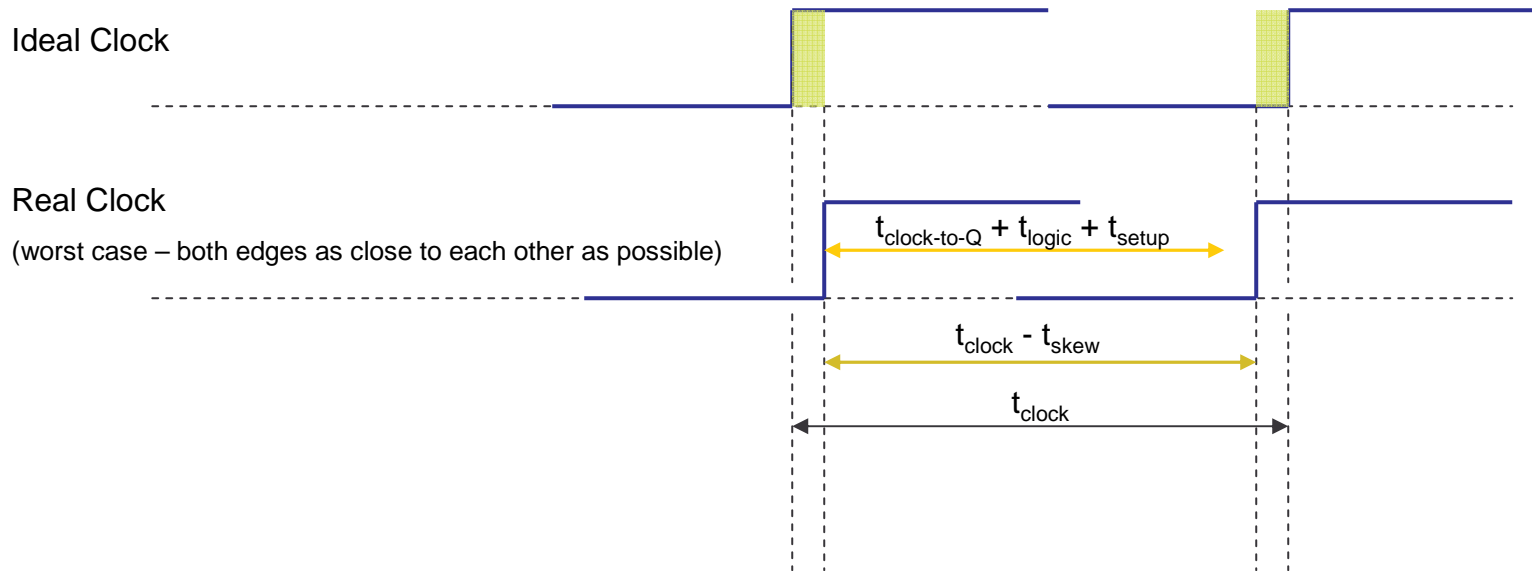
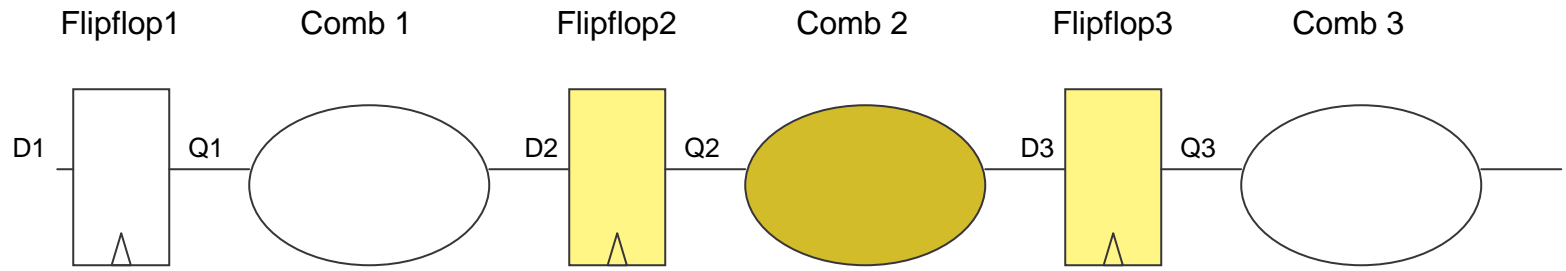
The flipflops do their job correctly only if the signal at their inputs does not change for some time before the clock edge ( $t_{\text{setup}}$ ) and some time after the clock edge ( $t_{\text{hold}}$ ).



In addition, the clock signal, which flows around the system on a separate set of wires called the clock tree, has its own variability, called *skew*.



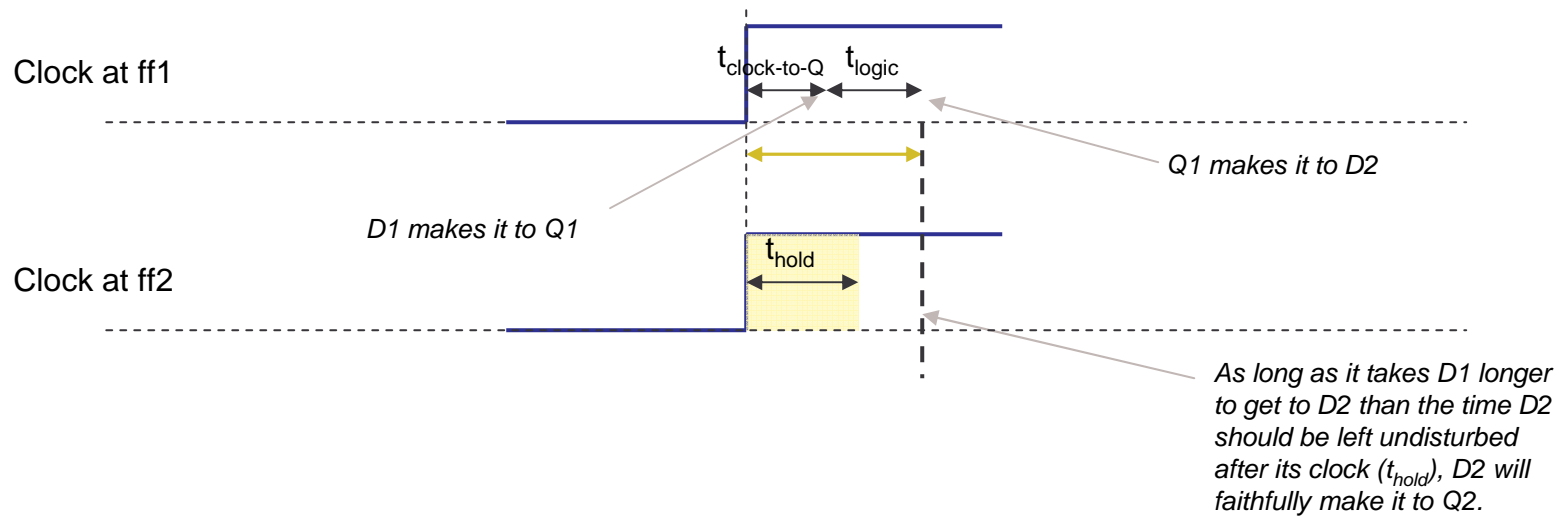
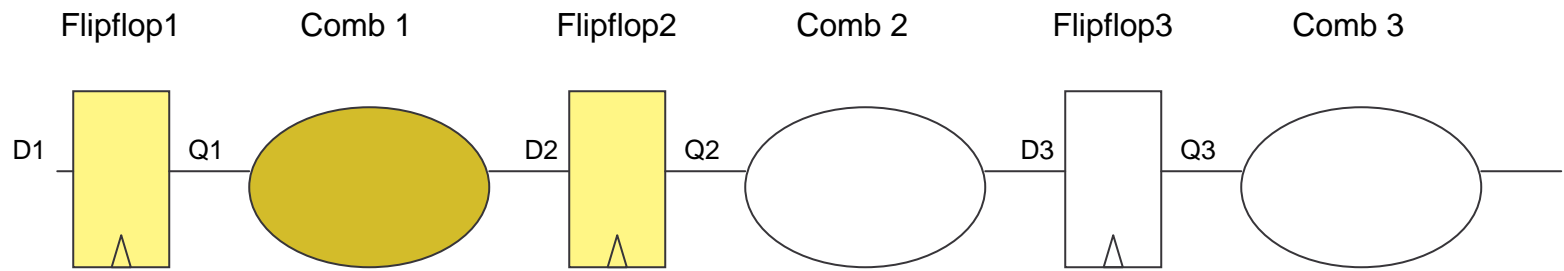
Time to propagate a valid (no violations) signal at D2, to D3, counting from the clock edge at Flipflop2, is invariably =  $t_{\text{clock-to-Q}} + t_{\text{logic}}$ . And for Flipflop3 to latch it, this signal has to be maintained at D3 for  $t_{\text{setup}}$  time before the clock tree sends the next positive edge of the clock to Flipflop3.



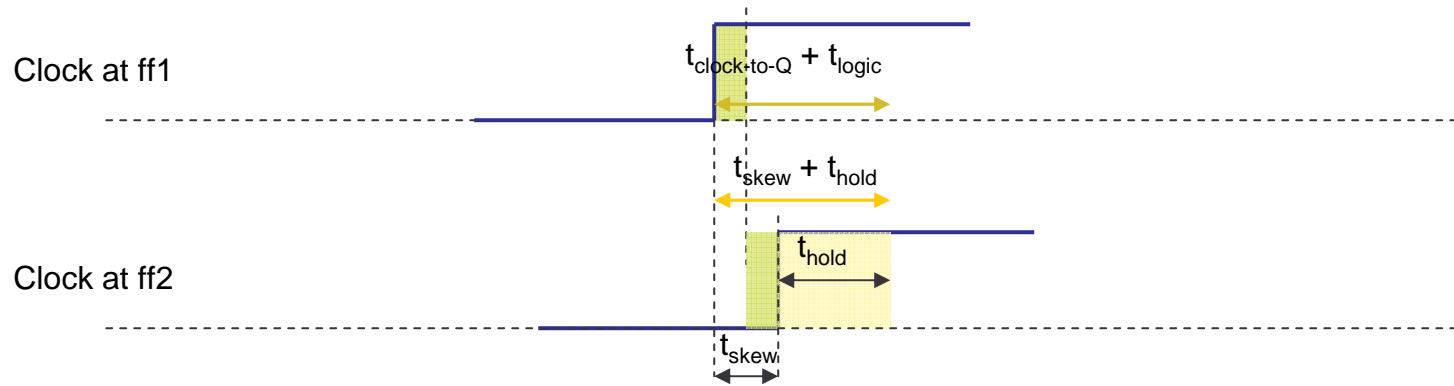
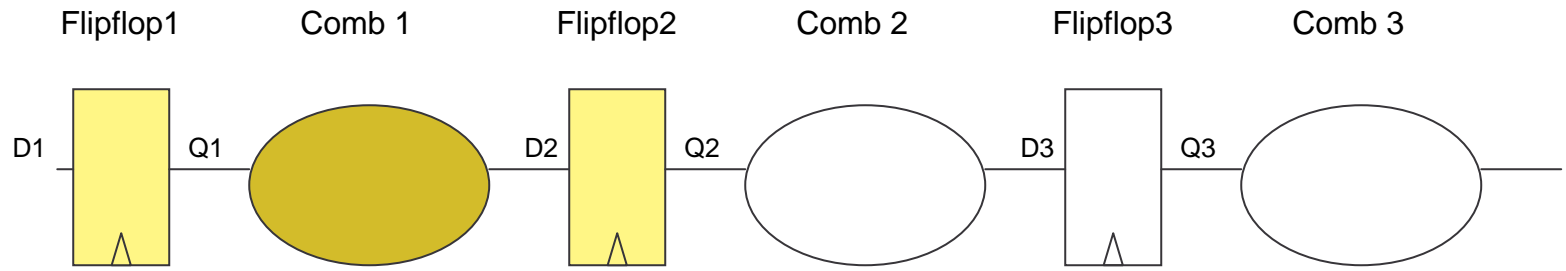
Therefore, to make sure the signal is SETUP at the input of Flipflop3 far enough ahead of the clock edge ...

That is, to prevent setup violations ...

$$\begin{array}{ccc}
 \leftarrow t_{\text{clock-to-Q}} + t_{\text{logic}} + t_{\text{setup}} \rightarrow & < & \leftarrow t_{\text{clock}} - t_{\text{skew}} \rightarrow \\
 & = &
 \end{array}$$



For  $D2$  to be able to send its signal to  $Q2$ , it must be left unchanged for  $t_{hold}$  time after a clock edge. That is, during this time, a signal from  $D1$  should not be able to race through the combinational logic Comb1 and make it to  $D2$ .



Therefore, to make sure the signal is HELD properly at the input of Flipflop2 without the input of the previous Flipflop (D1) racing through ...

That is, to prevent hold violations ...

$$\begin{array}{ccc}
 \leftarrow t_{\text{skew}} + t_{\text{hold}} \rightarrow & < & \leftarrow t_{\text{clock-to-Q}} + t_{\text{logic}} \rightarrow \\
 & = & \\
 \leftarrow t_{\text{skew}} \rightarrow & & \leftarrow t_{\text{hold}} \rightarrow
 \end{array}$$

# Summary

*In general, an inequality should still be satisfied if you use the largest possible values on the side that is supposed to be smaller, and the smallest possible values on the side that is supposed to be larger.*

to prevent setup violations ...

$$t_{\text{clock-to-Q\_max}} + t_{\text{logic\_max}} + t_{\text{setup\_max}} < t_{\text{clock}} - t_{\text{skew}}$$

The diagram shows the inequality  $t_{\text{clock-to-Q\_max}} + t_{\text{logic\_max}} + t_{\text{setup\_max}} < t_{\text{clock}} - t_{\text{skew}}$ . Below the left side of the inequality, a yellow double-headed arrow spans the width of the expression  $t_{\text{clock-to-Q\_max}} + t_{\text{logic\_max}} + t_{\text{setup\_max}}$ . Below the right side, another yellow double-headed arrow spans the width of the expression  $t_{\text{clock}} - t_{\text{skew}}$ . An equals sign is positioned below the less-than sign.

to prevent hold violations ...

$$t_{\text{skew}} + t_{\text{hold\_max}} < t_{\text{clock-to-Q\_min}} + t_{\text{logic\_min}}$$

The diagram shows the inequality  $t_{\text{skew}} + t_{\text{hold\_max}} < t_{\text{clock-to-Q\_min}} + t_{\text{logic\_min}}$ . Below the left side of the inequality, a yellow double-headed arrow spans the width of the expression  $t_{\text{skew}} + t_{\text{hold\_max}}$ . Below the right side, another yellow double-headed arrow spans the width of the expression  $t_{\text{clock-to-Q\_min}} + t_{\text{logic\_min}}$ . An equals sign is positioned below the less-than sign.