

---

# TECHNICAL NOTE

## FALSE CLOCKING OF CLOCKED FLIP-FLOPS

---

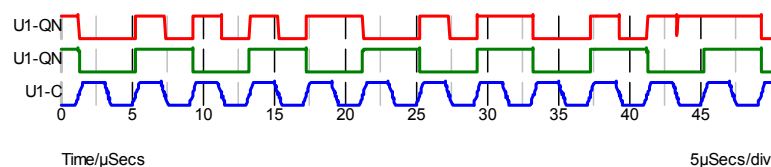
### OVERVIEW

This technical note explains a problem that occurs when simulating clocked flip-flops. The problem applies to all simulators not just SIMetrix.

In a nutshell, clocked flip-flops sometimes do not clock correctly if the time step gets too large. SIMetrix has an error control algorithm that overcomes this but it needs to be enabled with some option settings. The cause of the problem and how to use the option settings is described in this note.

### THE PROBLEM

The following graph show an example of the problem:



The waveforms are from the simulation of a D-type flip-flop configured as a divide by 2 counter. The blue trace is the clock and the green trace shows the correct result. The red trace is incorrect and is an example of the problem described in this note.

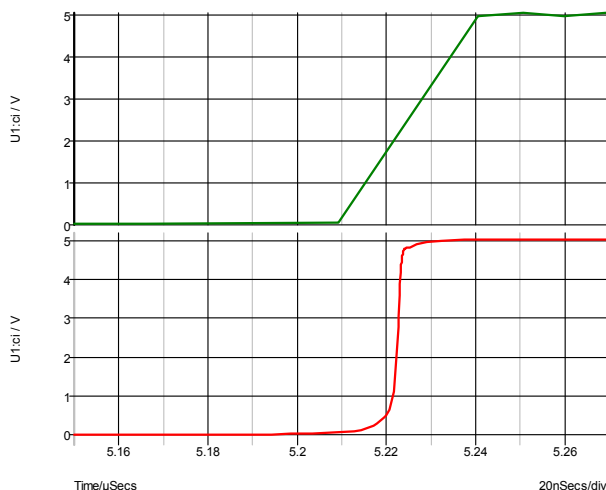
### THE CAUSE

The cause of the problem is related to a similar real-life problem whereby flip-flops fail to clock correctly when the clock edges are too long. This problem arises because all clocked flip-flops rely on a correct internal sequence of events. That sequence of events is governed by the delays inherent within the device but these delays are small and require the clock edge to be of a similar magnitude. If the clock edge is very slow, the threshold voltages of the internal gates become the dominant parameter in determining the internal sequence of events. In some situations, it is possible that the order of events changes completely when the clock edge is slow and this results in clocking errors.

The problem for simulation is similar but not identical. It occurs when the clock changes state in a single time step which is considerably longer than the edge speed required for correct clocking. For the simulation to proceed correctly, the internal gate delays must be simulated correctly for the reasons explained above. But, with a long time step, these delays will be insignificant and won't contribute significantly to the behaviour of the device. In effect, the simulation at the end of the clock edge will be similar to performing a static DC operating point. But we are simulating a bi-stable device and DC operating point simulations can arrive at either stable state. DC sweep analyses of bi-stable or semi-stable circuits (e.g. schmitt triggers) often don't behave as desired for this reason.

What is needed is a means of keeping the time steps small during the simulation of the rising edge. Unfortunately, the standard SPICE algorithms do not provide for the scenario that we have described. The time step is controlled by a number of factors, but the most important one is what is known as 'Local Truncation Error' control or LTE for short. This is a mathematical algorithm that attempts to estimate the error of the approximation used to simulate capacitors and inductors. This error is dependent on the time step; the smaller the time step the smaller the error. By estimating the error and comparing it with the tolerance parameters, the time step can be controlled to retain acceptable accuracy.

The algorithm is effective in keeping time steps for clocked flip-flops small during the rising edge as long as the simulator 'catches' the edge in the first place. This is illustrated below:



The green trace is the clock during the simulation that gave the wrong result. As can be seen, the clock jumped straight past the point where the edge should have been. The red trace shows the correct result and was obtained by simulating the circuit with an additional error control mechanism that allows the edge to be caught. Once the edge is caught, LTE takes over.

## THE SOLUTION

SIMatrix versions 4.5b and later have an additional - quite simple - error control mechanism that prevents the problem described here from occurring. This mechanism puts a limit on the amount by which, the voltage on any circuit node can change from one time step to the next. If the limit is exceeded, the time step is reduced until the change lies within a pre-defined tolerance. But note, that this error control mechanism is not enabled by default as it can slow down the simulation of circuits that do not use clocked flip-flops albeit only slightly (we estimate about 5-10%).

This error control mechanism is enabled by setting the option 'maxVDeltaRel'. Add this to your netlist or in the schematic's F11 window:

```
.OPTIONS maxVDeltaRel=0.4
```

We have found that a value of 0.4 works well. What this does is limit the maximum change in voltage on a node to 0.4 times the maximum voltage seen so far on that node. In the above example of a 5V circuit, this is 2V, but in fact the limit will be 2.5V as there is another option - maxVDeltaAbs - which adds an absolute value to this limit. maxVDeltaAbs has a default value of 0.5V.