A Methodology for the Offset-Simulation of Comparators

Achim Graupner
ZMD AG, Dresden

Version 1, October 1, 2006

A methodology for determining the input referred offset voltage of comparators is presented. This in general is difficult as the output of a comparator is discrete valued. The method relies on a Monte-Carlo-Simulation with certain comparator input values and some postprocessing of the comparator output data. The comparator is always operated in its intended environment, there is no modification of the comparator itself nor some unusual stimuli required. There is also no known restriction for the type of comparators to be analyzed.

Last updated on November 2, 2006. You can find the most recent version at www.designers-guide.org. Contact the author via e-mail at graupner@zmd.de.

1 Introduction

Comparators are a frequently used building block in analog circuit design. One of its most important properties is its input referred offset. For the most simple implementation, an amplifier with a high gain, this is a simple task that can be carried out with the help of a feedback network and some Monte-Carlo simulation.

However, most circuit implementations of comparators are much more complicated. Comparators often employ some hysteresis or some clever clocking scheme to reduce power dissipation or offset. In those cases however the determination of the input referred offset is no longer simple at all. In the latter case DC simulations are no longer sufficiently as the operation depends on the circuit's transient behavior. In the case of hysteresis a clever DC analysis has to be carried out because the circuit provides multiple operating points.

An method for determining the offset is described in [1]. Its basic idea is to build a first-order sigma-delta-modulator comprising the comparator and an ideal integrator. When steady state is reached, i.e. the high and low times of the comparator output are equal, then the mean input value equals the sum of the comparators threshold value plus the comparators offset. The method is quite simple but to achieve a meaningful accuracy the simulation time has to be considerably long until sufficient settlement. If the comparator employs a hysteresis then this method does not compute the two switching levels and their offset but a mean threshold value.

This short paper presents a very simple methodology to determine the offset of arbitrary comparators with the help of transient simulations. It requires a very simple testbench and some simple postprocessing only. In the following the method is introduced and illustrated with the help of an example.

2 Simulation Setup

For the proposed method a testbench of the comparators with is normal surroundings as supply and clock is used. Additionally a dedicated input has to be provided. The input value of the comparator has to be ladder-shaped, see Figure 1. This signal can be generated with a triangular shaped pulse followed by some ideal sample-an-hold or with a dedicated block in an analog hardware description language (AHDL). For each input value x_i the comparator is activated and its output value y_i is stored. For an ideal comparator for all input values x_i below the threshold value x_{th} the output value y_i is 0 for all values greater it is 1. This setup is shown in Figure 1. Taking device parameter mismatch into account this behavior might randomly change. This means the comparator output might be 0 even if $x_i > x_{\text{th}}$ or may be 1 although $x_i < x_{\text{th}}$.

The behavior of the comparator eventually depends on the actual device parameters, especially their matching. This effect can be modeled with an error signal x_{off} with is added to the comparator's input with all device parameters of the comparator at their nominal value, see Figure 2.

To evaluate the influence of the random device parameter variation Monte-Carlo-analysis is used. The result of each Monte-Carlo-iteration is collected. Now for all Monte-Carlo-samples for each input value the probability

FIGURE 1 *Stimuli for offset-simulation of a comparator.*

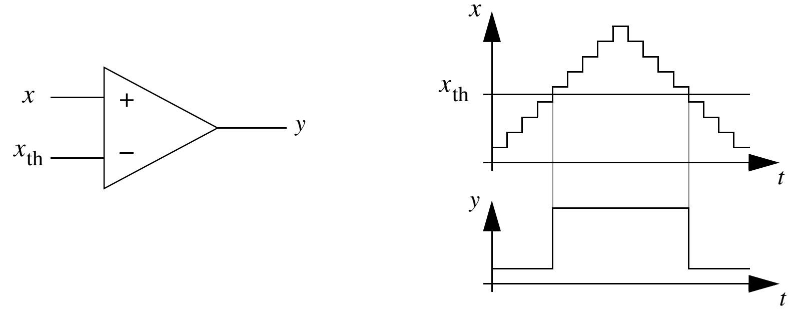
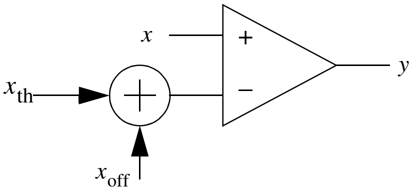


FIGURE 2 *Offset model of a comparator.*



$$\begin{aligned} P(y_i = 1) &= P((x_i - x_{th}) > x_{off}) \\ &= \frac{n_i}{N} = z_i \end{aligned} \tag{1}$$

can be estimated, where N is the number of Monte-Carlo-iterations and n_i is the number of runs where the comparator output is 1 when x_i is applied. This function is the integral of x_{off}'s probability density function. Hence from (1) the statistical properties from x_{off} can be easily computed. In order to do so (1) can be plotted in a normal probability plot. This is a graphical technique for assessing whether or not a data set is approximately normally distributed. The data x_{off} are plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line. From this line the mean value and the standard deviation can be determined.

In order to generate a normal probability plot the inverse of the cumulative normal distribution function is applied to the z_i

$$v_i = \Phi^{-1}(z_i) \tag{2}$$

where

$$\begin{aligned} \Phi(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(\frac{-t^2}{2}\right) dt \\ &= \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \end{aligned} . \tag{3}$$

Now a first order polynomial $z = p_1 x + p_2$ is fitted through the x_i vs. z_i plot. From the polynomials coefficients the mean value and the standard deviation of the distribution can be calculated:

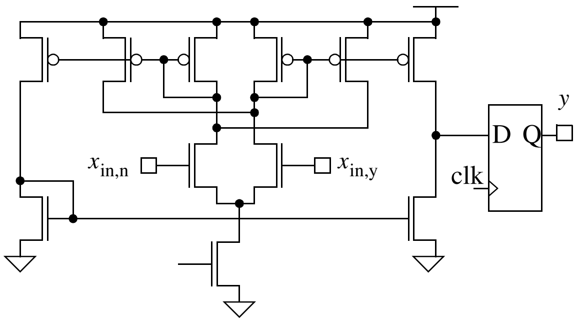
$$\mu_{\mathrm{xoff}} = \frac{-p_2}{p_1} - x_{\mathrm{th}}$$
$$\sigma_{\mathrm{xoff}} = \frac{1}{p_1} \tag{4}$$

This method is illustrated in the following example.

3 Example

The Figure 3 shows the functional principle of a non-clocked comparator incorporating hysteresis and a output latch.

FIGURE 3 *Simple comparator with hysteresis and output register.*



The result of a 700-runs Monte-Carlo simulation is shown in Figure 4. At the input a ramp with 1mV steps is applied. For each step of the ramp a comparison cycle is carried out and the comparator's result is stored (in our case into a file with means of an AHDL-block). In a subsequent postprocessing for each ramp value the number of comparator output 1 is counted and normalized to the number of runs.

It can seen both the hysteresis of the comparator as well as its sensitivity to device parameter mismatch. For the rising slope and an input voltage of 1.64V in about 60% of all simulations the comparator output had an output of 1, in the rest it remained 0.

The simulation data of the rising slope have been plotted in a normal distribution plot, see Figure 5. It is seen that the result is to a good approximation a straight line. Hence, the offset of the comparator is normal distributed and its mean value as well its standard deviation can be calculated using (4).

When normalizing the y-axis to a cumulative normal distribution then the statistical properties of the comparator's threshold value can be easily read out from the diagram (16%, 50% or 84%-values respectively), see Figure 6. In this case $\overline{(x_{\mathrm{off}} + x_{\mathrm{th}})} = 1.64$ and $\sigma_{\mathrm{voff}} = 0.018$.

In order to obtain reliable results the values for the x_i have to be selected carefully. The difference between two values should be less than the standard deviation of the offset.

FIGURE 4 *Result of a Monte-Carlo-simulation (cumulative histogram).*

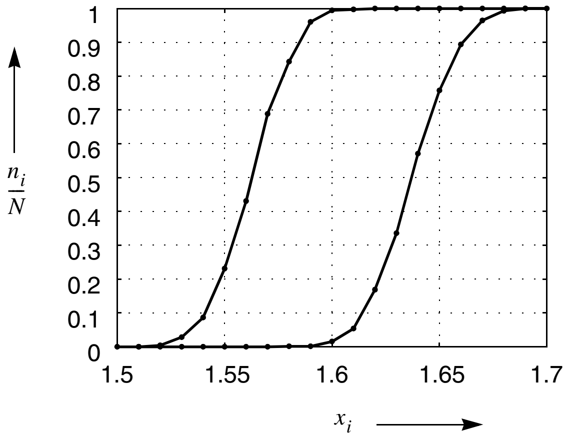


FIGURE 5 *Normal probability plot.*

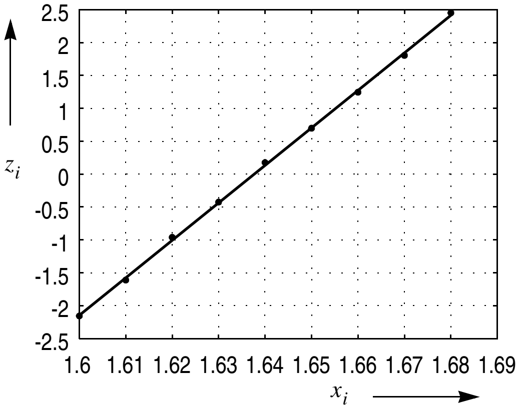
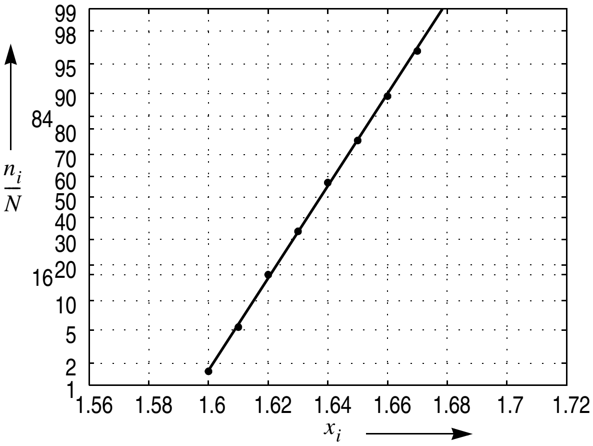


FIGURE 6 *Normalized cumulative frequency plot.*



Also the minimum and maximum values of x_i should always include the respective comparator threshold, so the interval should span several times the standard deviation around the comparator threshold. Those numbers have to be roughly guessed before starting the simulation.

4 Summary

A very simple methodology to determine the offset of arbitrary comparators with the help of transient simulations has been presented. It requires a very simple testbench and some simple postprocessing only. The comparator can be simulated in its intended operating regime.

4.1 If You Have Questions

If you have questions about what you have just read, feel free to post them on the *Forum* section of *The Designer's Guide Community* website. Do so by going to *www.designers-guide.org/Forum*.

Appendix

The algorithm has been implemented in the Matlab/Octave code shown in Listing 1.

LISTING 1 *Matlab code that implements the post-analysis portion of the algorithm.*

```
% it is assumed that the simulation results are provided
% in the matrix "ydata" and the array "xdata"

% (1) ni/N, Figure 2
y = sum(ydata)/length(ydata);
plot(xdata, y)

% (2) inverse erf for rising slope only
% y(1:21) is this case contains the data
% for the rising slope
v = sqrt(2)*erfinv(y(1:21)*2–1);

% select values within –2.5 ... 2.5 sigma
i=find(v>–2.5 & v<2.5)

% fit line through selected values
% Figure 5
[p,s]=polyfit(xdata(i), v(i), 1);
plot(xdata(i), v(i), xdata(i), polyval(p,x(i)))

% (4) compute mu and sigma
sigma = 1/p(1);
mu = –p(2)/p(1);
```

References

[1] T.W. Matthews and P.L. Heedley. A simulation method for accurately determining DC and dynamic offsets in comparators. 48th Midwest Symposium on Circuits and Systems, pp. 1815-1818, vol. 2, Aug. 2005.