7.1 Characterizing PM Jitter

SpectreRF's PNoise analysis computes the time-average power spectral density of the noise at the output of the block. If this noise is stationary (as opposed to cyclostationary), it is a simple matter to apply (31) to calculate the jitter. Simply choose a representative set of periodic inputs to the block and use SpectreRF's Periodic Steady State (PSS) analysis to compute the steady-state response. This computes the periodic operating point about which the noise analysis is performed. It also gives $dv(t_c)/dt$, the slew rate of the output at threshold crossing. Apply SpectreRF's PNoise analysis to compute the noise power at the output as a function of frequency. Choose the frequency range of the analysis so that the total noise at frequencies outside the range is negligible. Thus, the noise should be at least 40 dB down and dropping at the highest frequency simulated. Finally, integrate the noise power over frequency and apply Wiener-Khinchin Theorem [16] to determine

$$\mathrm{var}(n_v) = \int_{-\infty}^{\infty} S_{n_v}(f)df \ , \tag{57}$$

the total noise power [9], and apply (31).

In general, the noise is strongly cyclostationary and so the above procedure is insufficient. When the noise is cyclostationary the same procedure is used, except a gating function is applied to the output so that only the noise that occurs near the threshold crossing is considered in the jitter calculation. SpectreRF's PNoise analysis computes the time-average of the noise at the output and it is not possible in general to post-process the PNoise results to determine the noise at the time of the threshold crossing. Rather, a limiter is added to the output of the block and SpectreRF computes the noise at the output of the limiter. The limiter, given in Listing 9, is designed to saturate when the output of the block is outside a certain range to prevent any noise at the output from being considered except the noise present near when the signal crosses the threshold, as shown in Figure 5. The range of the limiter, V_L and V_H, is chosen such that the noise and the slew rate is approximately constant while the limiter is active. When running PNoise analysis, assure that the maxsidebands parameter is at least ten times larger than T/t_{ti} for any i. This assures that the narrow noise pulses are adequately resolved by the PNoise analysis. Jitter is independent of T, so to reduce the number of sidebands needed, use T as small as possible. A symptom of maxsidebands not being set sufficiently large is that the extracted value of jitter increases as T decreases.

Assume that the amplitude of the noise during each transition is the same, then

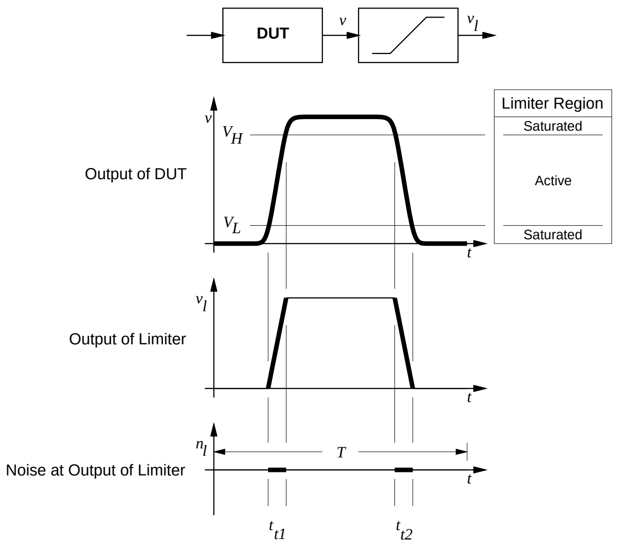
$$\mathrm{var}(n_l, t_c) \cong \frac{\langle n_l^2 \rangle T}{t_{t1} + t_{t2}} \tag{58}$$

where $\langle n_l^2 \rangle$ is the time average of the noise power at the output of the limiter. SpectreRF computes the power spectral density of the noise at the output of the limiter, and by the Parseval's Theorem,

$$\langle n_l^2 \rangle = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} n_l^2(t)dt = \int_{-\infty}^{\infty} S_{n_l}(f)df \ . \tag{59}$$

If we further assume that $t_t = t_{t1} = t_{t2}$, then

FIGURE 5. *When the device-under-test (DUT) exhibits cyclostationary noise, a limiter is applied to output of driven block to suppress noise present outside of the active transitions. $v(t)$ is the output of the block, $v_l(t)$ is the output of the limiter, and $n_l(t)$ is the noise at the output of the limiter. Noise on a waveform is denoted by using a thick trace. The output of the limiter is only noisy when it is inside its active region (when it is not limiting).*



LISTING 9. *Limiter used to characterize PM jitter in binary signals.*

```
// Simple Limiter

‘include “discipline.h”
‘include “constants.h”

module limiter (out, in);

output out; input in; electrical out, in;

parameter real Vlo=–1, Vhi=1;

analog begin
    // Place time-point at threshold crossings
    @(cross(V(in) – Vlo) or cross(V(in) – Vhi));

    // Determine the output
    if (V(in) < Vlo)
        V(out) <+ Vlo;
    else if (V(in) > Vhi)
        V(out) <+ Vhi;
    else
        V(out) <+ V(in);
end
endmodule
```

$$\dot{v}(t_c) \approx \frac{V_H - V_L}{t_t} . \qquad (60)$$

And from (32) and (33)

$$J = \frac{\sqrt{2\operatorname{var}(n_l, t_c)}}{\dot{v}(t_c)} \approx \sqrt{\frac{2\langle n_l^2\rangle T}{Kt_t}} \frac{t_t}{V_H - V_L} , \qquad (61)$$

$$J \approx \frac{\sqrt{2\langle n_l^2\rangle T t_t / K}}{V_H - V_L} , \qquad (62)$$

where K is the number of transitions that occurred during the period. If $K = 2$,

$$J \approx \frac{\sqrt{\langle n_l^2\rangle T t_t}}{V_H - V_L} . \qquad (63)$$

This general methodology for characterizing the PM jitter of driven blocks with binary outputs is extended or clarified for important special cases in the next few sections.

7.1.1 PM jitter of the PFD/CP

The PFD and CP work together to generate a three-level discrete-valued signal (it takes the values –1, 0, and +1) whose time average is used as the loop error signal. The average of this signal controls the VCO after it has been extracted by the LF.

There are two aspects of the PFD/CP that differ from the assumptions made above. First, the output of the CP is a current, so the limiter and the equations given in the previous section need to be adapted. Second, the output of the CP has three distinct levels rather than the two assumed above. Thus, the CP has a ternary output rather than a binary output.

The first step in predicting the jitter of the PFD/CP is to get the PFD/CP to generate jitter. To do so, drive the PFD with two signals at the same frequency but offset in phase so that there is a periodic pulse train at the output of the CP. In this case, the output of the CP has two levels, so you can use the limiter given Listing 9 if you use a CCVS to convert the output current to a voltage. This allows you to compute the jitter for this particular set of two levels. You could then reverse the phase difference of the two input signals to compute the jitter for the other set of two levels. So if the first time you characterized the jitter when the output is transitioning between 0 and 1, then with the second time you are characterizing the jitter when the output is transitioning between 0 and –1. You will have to reverse the input on the CCVS during the second measurement. This gives you two different jitter values, use the worst.

7.1.2 PM Jitter of a FD

With ripple counters, one usually only characterizes one stage at a time. The total jitter for the ripple counter is then computed by taking the square-root of the sum of the square of the jitter on each stage.