Analysis of Random Jitter in a Clock Multiplying DLL Architecture

R.C.H. van de Beek[1], E.A.M. Klumperink[1], C.S. Vaucher[2] and B. Nauta[1]

[1] University of Twente
Dept. of Electrical Engineering, IC-Design group
P.O. Box 217, 7500 AE Enschede, the Netherlands
Tel. +31 (0)53 489 2644 Fax. +31 (0)53 489 1034
E-mail: r.c.h.vandebeek@el.utwente.nl

[2] Philips Research Laboratories
5656 AA Eindhoven, The Netherlands

Abstract— **In this paper, a thorough analysis of the jitter behavior of a Delay Locked Loop (DLL) based clock multiplying architecture is presented. The noise sources that are included in the analysis are the noise of the delay elements, the reference jitter and the noise of the Phase Frequency Detector and Charge Pump combination. It is shown that the effect of all noise sources on the output timing jitter can be minimized by minimizing the loop gain of the DLL. This means that the loop is merely used to tune the delay of the Delay Line to a nominal value of exactly one reference input period; the loop is ineffective in filtering jitter. The analysis results are verified using high-level simulations, with good agreement.**

I. Introduction

A very important building block in almost all digital and mixed signal Integrated Circuits is the clock multiplier. This building block multiplies the incoming reference clock frequency by a certain factor. An important reason for clock multiplication is often that no crystals are available with a clock frequency as high as needed on-chip. Another application of clock multiplication lies in the fact that it often saves power to transport the clock from chip to chip at a frequency that is lower than the on-chip clock frequency, making clock multiplication at the receiving end necessary. Also, when parallel data is to be serialized using a multiplexer, clock multiplication is needed to time the outgoing bits. In all these applications, the quality of the multiplied clock with respect to timing jitter is an important specification.

An often encountered and well-known architecture used to realize the clock multiplier is the Integer-N PLL. In this architecture, a Voltage Controlled Oscillator (VCO) is locked to a clean reference clock. A frequency divider in the feedback loop is used to make the VCO run at a frequency that is an integer multiple of the reference frequency.

Recently, architectures based on a Delay Locked Loop (DLL) have been successfully used as Clock Multipliers [1][2]. In this architecture, which is schematically shown in Figure 1, a Voltage Controlled Delay Line (VCDL) instead of a VCO is locked to the clean reference. The extra timing information needed to generate the high frequency clock is now obtained by using a Delay Line that consists of multiple tuneable delay elements, thus generating multiple phases of the low frequency clock. These phases are then combined into one high frequency clock using a circuit that is referred to as Edge Combiner. Note that this technique is only applicable if the multiplication ratio is a constant integer number. The advantage of the DLL based architecture is that the Delay Line is 'reset' with respect to jitter every time a new reference edge is applied at the input, whereas in the VCO of a PLL, the jitter accumulates [3].

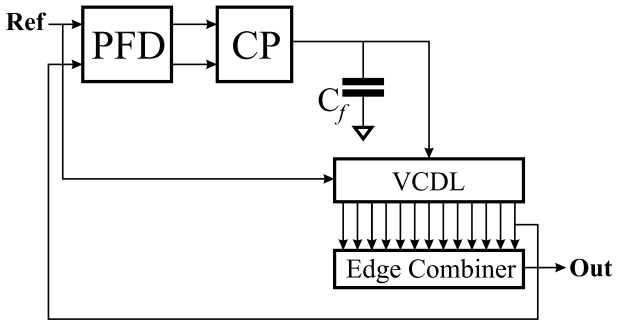

Fig. 1. The DLL-based architecture

Because of the low jitter that is demanded of the multiplied clock, it is important to be able to calculate the expected jitter of the DLL architecture as a function of DLL design parameters. In [3] a design equation with respect to DLL jitter is derived. The only source of jitter that is

taken into account, though, is the jitter that is generated in the Delay Line of the DLL. In practice, there are more jitter sources. The loop components other than the Delay Line will also be influenced by noise, which is reflected through the loop to the output of the DLL. Also, the jitter that is present on the reference clock will be measurable as DLL output jitter.

In this paper, the timing jitter of a DLL is analyzed, also taking into account the other before mentioned noise sources in order to offer a more complete set of design equations from which the DLL output jitter can be predicted. This is done by first composing a mathematical model of difference equations describing the DLL behavior. The output jitter due to different noise sources is then analyzed in the time domain directly.

In section II of this paper, the DLL architecture and the Edge Combination process are described briefly. A mathematical model of the DLL is proposed in section III, which is then used to calculate the effect of the Delay Line noise, the PFD/CP noise and the reference jitter on the DLL output jitter. To verify these results, a DLL simulation model is shown in section IV, along with results of performed simulations. The paper concludes in Section V with a summary of the results.

II. The DLL architecture

Figure 1 shows the general architecture of a DLL with edge combiner. The loop itself consists of a Phase Frequency Detector (PFD) that is combined with a Charge Pump (CP). The loop filter consists of a simple capacitor that integrates the charge pulses coming from the CP. In a PLL such a simple filter would lead to stability problems because of the integrating function of the VCO used in a PLL; in a DLL however there is no pure integrator other than the CP combined with the loop filter capacitor, which guarantees stability.

The basic idea behind a DLL-based clock multiplier is that the reference input is delayed using a Delay Line with multiple output taps, as shown in Figure 3. The total delay of the line is controlled by the loop to be equal to the input period of the reference clock. The different output taps now deliver different phases of a clock that has a frequency equal to the input reference. These extra clock phases contain extra timing information that can be combined into one clock with a frequency that is an integer multiple of that of the reference clock. This has been illustrated in Figure 2.

If only the rising edges of the different clock phases are used to generate both the rising and falling edges of the generated clock, it is easy to show that the number of output taps needed, is equal to twice the frequency multiplication factor. In some cases it is possible to also use the falling edges of the different clock phases to generate timing information. This will however cause timing information that is dependent on the duty cycle of the reference, which might form a problem in some applications.

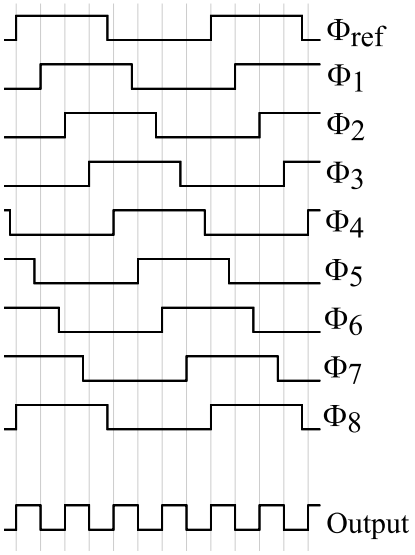

Fig. 2. The edge combination process for $N = 4$, using only rising edges to generate the output clock

It is also possible to generate the rising edges of the output signal directly from the rising edges of the different clock phases, while the falling edges of the output signal are generated by the use of a resonator, as described in [1]. A disadvantage of this method is that an inductor is used, which consumes area and is more difficult to port to newer technologies than a purely digital solution.

In this paper, we assume that only the rising edges of the different clock phases are used without a resonator (Figure 2 being an example of this), and thus the number of delay cells M in the Delay Line equals:

$$M = 2N \tag{1}$$

where N is the ratio between the output frequency of the edge combiner and the incoming reference frequency.

III. The DLL Jitter Analysis

This section describes the actual analysis of the DLL Jitter. To calculate the effect of noise sources in the different building blocks of the DLL architecture, the loop behavior has to be taken into account. This is done here using difference equations describing the loop mathematically. These difference equations are then used to calculate the variance of the jitter in the time domain directly. First, the DLL output jitter due to the internal Delay Line noise is analyzed. Then, the jitter resulting from the circuit noise of the Phase

Frequency Detector and the Charge Pump is analyzed. Finally, the jitter originating from a polluted reference signal source is calculated.

A. *Mathematical model of the DLL with Noisy Building Blocks*

First, a set of difference equations describing the DLL behavior is derived. This equation set is then used to analyze the jitter originating from the different noise sources of the DLL and the reference signal source.

To be able to calculate the 'jitter', first a quantitative definition of jitter is needed. There are many different definitions for jitter available in literature. In this work, a very simple and intuitive definition will be used:
Jitter is the random deviation in time of the zero-crossings of a certain generated clock with respect to corresponding zero-crossings of an *ideal* clock. The ideal clock has zero-crossings that are separated by a constant amount of time which equals the mean period of the generated clock.

For the stochastic DLL jitter analysis, the model shown in Figure 3 is used. Naturally, the 'ideal clock' is no part of the actual DLL; it is merely being shown to illustrate the concept of jitter that is being used here.

The Phase Frequency Detector compares the zero-crossing times of the reference to those of the last tap of the Delay Line. It is assumed is that the loop has successfully locked to the state in which the Delay Line delays the input signal by one period time of the reference clock. The Charge Pump converts the measured time difference into a charge which is pumped into the loop filter (a simple capacitor), which integrates this charge. The quantity $q\,(n)$ denotes this amount of charge. Note that parameter n is used to indicate the period number of the input clock; this variable is used in the difference equations that are derived shortly.

The tuning voltage v_c determines the delay of the Delay Line d_{tot} according to:

$$d_{tot} = T_0 - K_d v_c + \Delta d_{tot} \tag{2}$$

where T_0 is the delay of the Delay Line for a tuning voltage of zero volts and Δd_{tot} is the jitter added by the Delay Line. For convenience, the assumption is made that T_0 equals the period time of the clock T_S. This simplifies analysis, while the results of the jitter calculations do not depend on the value of T_0 (it is comparable to a DC biasing point).

Deviations in the tuning voltage, as well as jitter added by the delay cells will result in jitter on the taps of the Delay Line. Also jitter present on the reference clock that is fed into the Delay Line is directly measurable as jitter

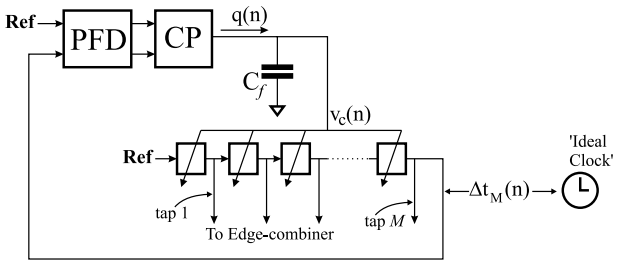

Fig. 3. The DLL-model that is used; the ideal clock illustrates the jitter definition used here

on the output taps. The effect of both the tuning voltage errors and the jitter added by the delay cells will be worst at the last output tap of the Delay Line, which means the jitter variance will be highest at the last output tap. This holds as long as we assume that the jitter contributed in a certain period of the input clock by a certain delay cell is neither correlated to that delivered by another delay cell, nor by previous jitter contributions of the same delay cell. We also assume that jitter added by the reference source is uncorrelated to the reference jitter history. Because we now know that the jitter variance will be highest at the last output tap, this is the only tap of which the jitter will be analyzed here.

The total amount of jitter at the last tap Δt_M can be expressed as:

$$\Delta t_M = -K_d v_c + \sum_{l=1}^{M} \Delta d_l + \Delta t_{ref} \tag{3}$$

where Δd_l is the jitter added by the l-th delay cell and M is the total number of delay cells. The term Δt_{ref} denotes the jitter on the input reference. This term appears because the jitter that is put in the Delay Line will appear on the output taps as well.

Now a set of difference equations describing the system of Figure 3 can be formulated:

$$q\,(n) = I_{CP}\left\{\Delta t_M\,(n) - \Delta t_{ref}\,(n) + \Delta t_{PFD}\,(n)\right\} + \\ + q_{noise}\,(n) \tag{4a}$$

$$v_c\,(n) = v_c\,(n-1) + \frac{q\,(n-1)}{C_f} \tag{4b}$$

$$\Delta t_M\,(n) = -K_d v_c\,(n) + \sum_{l=1}^{M} \Delta d_l\,(n) + \Delta t_{ref}\,(n-1) \tag{4c}$$

In these equations, $q\,(n)$ is the charge that the Charge Pump pumps into the loop filter after input period number n with q_{noise} denoting the part of that charge caused by a noisy Charge Pump, I_{CP} is the Charge Pump current,

$\Delta t_M(n)$ is the jitter of the last (M-th) output tap of the Delay Line after the n-th input period, $\Delta t_{PFD}(n)$ is the detection error that the Phase Frequency Detector makes due to its input referred voltage noise, which will be discussed in more detail later. The control voltage of the Delay Line during the n-th input period is denoted by $v_c(n)$.

This set of difference equations is used in the following, first to calculate the amount of jitter originating from the noise of the delay cells, then to calculate the jitter due to the noise in the Phase Frequency Detector and Charge Pump circuitry, and finally to analyze the effect of reference jitter.

B. DLL Output Jitter due to Delay Line Noise

In this section the jitter that will result at the different output taps of the Delay Line due to its own jitter is analyzed, using the set of difference equations (4). To isolate the effect of the delay cell noise, the other noise sources are neglected, assuming that the effect of these may be added later using superposition. This is allowed as long as the noise sources are not correlated and superposition holds, which seems reasonable as long as the jitter remains low. Also assumed is that the jitter contributed in a certain period of the input clock by a certain delay cell is neither correlated to that delivered by another delay cell, nor by previous jitter contributions of the same delay cell.

If (4a) is substituted in (4b), the set of difference equations can now be written as:

$$v_c(n) = v_c(n-1) + \frac{I_{CP}}{C_f}\Delta t_M(n-1) \tag{5a}$$

$$\Delta t_M(n) = -K_d v_c(n) + \sum_{l=1}^{M} \Delta d_l(n) \tag{5b}$$

neglecting all noise sources except the jitter caused by the delay cell noise.

Following the method described in the Appendix, we can find the variance of the signal Δt_M, which is the jitter variance of the last output tap of the DLL:

$$\sigma^2_{\Delta t_M} = \sigma^2_{\Delta d_l} \cdot M \frac{2}{2-\varepsilon} \tag{6}$$

with the so called *loop gain* ε defined as [3]:

$$\varepsilon \equiv \frac{I_{CP} K_d}{C_f} \tag{7}$$

Note that (6) is in agreement with the result achieved in [3].

It is important to note that the jitter is lowest for low values of the DLL loop gain ε, in which case the jitter would be equal to that of a Delay Line that is not controlled by a loop. This shows that the function of the control loop is not to remove jitter from the Delay Line but merely to tune the total delay of the Delay Line to the desired value.

This result can be intuitively explained by noting that for low frequency components of the delay cell noise, the loop indeed filters the jitter. For the high frequency components however, the delayed correction of the loop actually amplifies the noise energy via the Delay Line control voltage v_c. The output jitter would be lowest if no control action would be taken in that case, as it will be too late to correct for the high frequency part of the noise. According to (6), the net result is that the *total* jitter at the output will always be higher than the jitter of an uncontrolled Delay Line.

C. DLL Output Jitter due to PFD and CP noise

Apart from the jitter that is generated by the Delay Line, the loop components that take care of the feedback mechanism also introduce jitter. This is because the detection of the time difference between the reference signal and the Delay Line output signal is not perfect in practice.

First, the Phase Frequency Detector (PFD) that has to detect zero-crossings is realized using noisy elements. The internal noise of the Phase Frequency Detector can be calculated back to the input as a voltage noise, which influences the moment in time that the PFD generates its output signals and thus the charge that is integrated on the loop capacitor, if one assumes that the incoming edges are not infinitely steep.

Also, the Charge Pump (CP) generates jitter as the charge that is pumped into the loop capacitor is noisy, because the switched current sources inside the CP are noisy in a realistic implementation.

The jitter variance at the last output tap (numbered M) will be highest, as was the case for jitter caused by the Delay Line noise. This is because the jitter is caused by deviations in the control voltage of the Delay Line, which causes similar deviations in the delay of each delay cell. The effect of this is cumulative.

To simplify calculations, the CP noise is calculated back to the input of the Phase Frequency Detector as an equivalent time error:

$$\Delta t'_{PFD}(n) \equiv \Delta t_{PFD}(n) + \frac{q_{noise}(n)}{I_{CP}} \tag{8}$$

Assuming all noise sources absent except the PFD and CP noise and using the definition of $\Delta t'_{PFD}$ that is given in the previous equation, the set of difference equations (4) can be written as:

$$v_c(n) = v_c(n-1) + \frac{I_{CP}}{C_f}\Delta t_M(n-1) + \\ + \frac{I_{CP}}{C_f}\Delta t'_{PFD}(n-1) \quad (9a)$$

$$\Delta t_M(n) = -K_d v_c(n) \quad (9b)$$

Using a method similar to the calculation of the jitter due to Delay Line noise as described in the Appendix, the variance of the output jitter is calculated. This results in:

$$\sigma^2_{\Delta t_M} = \sigma^2_{\Delta t'_{PFD}} \frac{\varepsilon}{2-\varepsilon} \quad (10)$$

From this equation it is apparent that a small loop gain ε is beneficial for the DLL output jitter that is caused by the Phase Frequency Detector and the Charge Pump. The gain of the Delay Line should however be large enough to compensate for process spread and temperature variations; the Charge Pump current can not be chosen too small because of the jitter resulting from mismatch in the Charge Pump. This means that the loop filter capacitor should be made large at the cost of area, in order to maintain a reasonably low loop gain.

D. DLL Output Jitter due to Reference Jitter

Although it is a well-known fact that a DLL is not a suitable architecture when the reference signal that is applied to it has jitter of its own, as this is passed directly to the Delay Line output taps, it is still interesting to calculate the effect of reference jitter to the output, because it is realistic to assume that the reference signal actually does contain some jitter (*e.g.* because of clock buffering).

The loop will compare a certain reference edge with a delayed version of the previous reference edge. Assuming the reference jitter to be uncorrelated from period to period, this comparison will result in random charge pulses from the Charge Pump. The effect of these random charge pulses is a deviation of the control voltage of the Delay Line, which in turn causes the delay of the delay cells to deviate from their ideal values. The effect of this will again be highest at the last output tap of the Delay Line.

The effect of the reference jitter can be calculated using the set of difference equations (4) that describe the DLL behavior. The reference jitter is assumed to be the only present noise source. When other noise sources come into the picture, this can again be accounted for using superposition.

Assuming the reference jitter to be the only noise source yields the following difference equations:

$$v_c(n) = v_c(n-1) + \frac{I_{CP}}{C_f}\Delta t_M(n) - \frac{I_{CP}}{C_f}\Delta t_{REF}(n) \quad (11a)$$

$$\Delta t_M(n) = -K_d v_c(n) + \Delta t_{REF}(n-1) \quad (11b)$$

Using a method similar to the one described in the Appendix, the jitter on the last output tap due to the reference noise is analyzed, resulting in the following equation:

$$\sigma^2_{\Delta t_m} = \sigma^2_{\Delta t_{REF}} \cdot \left\{1 + 4\frac{\varepsilon}{2-\varepsilon}\right\} \quad (12)$$

Observing this equation shows that the jitter on the output taps is always more than the reference jitter, which shows, according to expectance, that a DLL is not suitable to filter out reference noise. This equation also shows that the lower the value of the loop gain ε, the lower the output jitter due to reference jitter.

IV. Simulation Results

To verify the jitter predictions that are described in the previous section, a high level simulation model has been used. This model is depicted in Figure 4. The three most important noise sources used in the analysis can be applied independently. The delay cell noise is modelled by random uncorrelated delay variations with zero mean. The Charge Pump noise is modelled by adding white noise to the Charge Pump current sources. The variance of the charge that is pumped into the filter is then roughly proportional to the PFD reset time (this is the overlap time of the up- and down-current sources that is present in realistic PFD designs [4]). The reference buffer that is used is comparable to the delay cells used in the Delay Line, *i.e.* it adds jitter to the reference signal that is uncorrelated from period to period.

To evaluate the simulated jitter, the clean positive zero crossings of the reference generator (before polluting it with jitter by the reference buffer) are compared with those of the DLL output. The jitter is then calculated as the variance of the time differences.

The graph shown in Figure 5 shows simulation results for certain values of the applied noise sources. The solid lines in this graph show the expected jitter using the equations that are derived in this paper.

First, simulations were done with only one noise source turned on with the variances of the other two put to zero. The graph shows a good agreement between the predicted and the simulated points. Then, all three noise sources were turned on to prove that the superposition principle, that was used as an important assumption throughout the analysis, was valid (meaning the jitter contribution of the

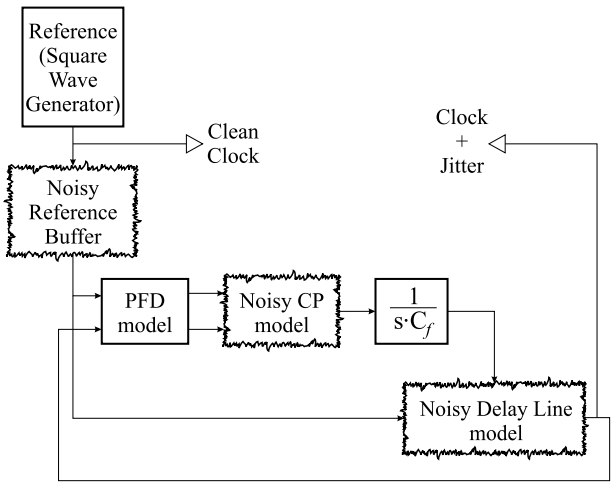

Fig. 4. The simulation model

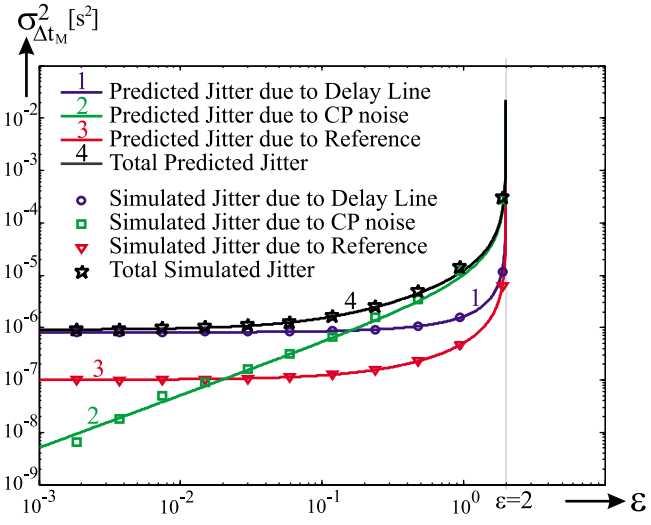

Fig. 5. The simulation results

different noise sources can indeed be added power-wise). The result of this simulation is also shown in Figure 5, again showing good agreement with expectations.

The simulation results give confidence in predictions of DLL output jitter that can be made using the equations derived in this paper.

V. Conclusions

The DLL jitter analysis has shown that the effect of all the noise sources in a DLL on the output jitter can be minimized by minimizing the loop gain of the DLL. This shows that the function of the control loop is not to filter out jitter (as is the case for a PLL), but merely to tune the value of the mean delay of the Delay Line to be equal to the reference period. For a very small loop gain, the DLL behaves as if uncontrolled with respect to jitter. The minimum DLL output jitter is then approximately equal to the jitter caused by the delay cell noise superposed on the jitter that is contaminating the reference signal. The jitter caused by the PFD/CP combination will be negligible for small values of the DLL loop gain. Note however that the minimum value of the loop gain is limited in practical designs. The Charge Pump current can not be taken arbitrarily small because of mismatch limitations. The gain of the controlled Delay Line should be large enough to enable tuning out process and temperature variations. This leaves the value of the loop filter capacitor as the parameter to minimize the loop gain, but this value will in practice be limited by area restrictions. Other practical issues such as settling behavior may also limit the minimum value of the loop gain.

Appendix

To demonstrate how to obtain the output jitter of a system described by difference equations, the calculations leading to (6) are shown in this Appendix. This equation can be derived using the set of difference equations given by (5), which describes the DLL behavior with the delay cell noise being the only source of jitter. The set of equations used is repeated here for convenience:

$$v_c(n) = v_c(n-1) + \frac{I_{CP}}{C_f}\Delta t_M(n-1) \tag{13a}$$

$$\Delta t_M(n) = -K_d v_c(n) + \sum_{l=1}^{M} \Delta d_l(n) \tag{13b}$$

The quantity of interest is the variance of the signal Δt_M. Because the mean of this signal is zero (as this is a linear system and the noise sources have zero mean), the variance of Δt_M can be written as:

$$\sigma^2_{\Delta t_M} = E\left\{\left(-K_d v_c(n) + \sum_{l=1}^{M}\Delta d_l(n)\right)^2\right\} =$$
$$= K_d^2 \cdot E\left(v_c^2(n)\right) - 2K_d \cdot E\left(v_c(n)\sum_{l=1}^{M}\Delta d_l(n)\right) +$$
$$+ E\left\{\left(\sum_{l=1}^{M}\Delta d_l(n)\right)^2\right\} \tag{14}$$

Because the variance of the tuning voltage does not depend on the period number n in the locked situation, this equation can be reduced to:

$$\sigma^2_{\Delta t_M} = E\left(\Delta t_M^2\right) = K_d^2 \cdot E\left(v_c^2\right) + M \cdot E\left(\Delta d_l^2\right) \tag{15a}$$

taking into account the variables in (14) that are uncorrelated.

This equation shows that in order to relate the variance of Δt_M directly to the delay cell noise variance, the variance of the tuning voltage v_c needs to be known. This variance can be found by using (13a). The following equation can be derived from it by taking the square on both the left and the right hand side, followed by equating the expected value of both sides, taking into account the uncorrelated variables:

$$E\left(v_c^2\right) = E\left(v_c^2\right) + 2\frac{I_{CP}}{C_f} E\left\{v_c\left(n\right)\Delta t_M\left(n\right)\right\} + \\ + \frac{I_{CP}^2}{C_f^2} E\left(\Delta t_M^2\right) \quad (15b)$$

Note that all expected values are independent of the value of n; if the equation still features this variable it is only to clarify the time relationship between two different variables.

Now there are two equations with three unknowns. To solve this problem, a new equation can be derived by adding $K_d v_c(n)$ on both sides of (13b), which seemingly does not give new information. Squaring this equation and equating the expected value of both the left and the right hand side, though, actually results in a new independent equation, making it possible to solve for the tuning voltage variance:

$$K_d^2 \cdot E\left(v_c^2\right) + 2K_d \cdot E\left\{v_c\left(n\right)\Delta t_M\left(n\right)\right\} + \\ + E\left(\Delta t_M^2\right) = M \cdot E\left(\Delta d_l^2\right) \quad (15c)$$

Finally, solving the set of equations (15) for $E\left(\Delta t_M^2\right)$ results in:

$$E\left(\Delta t_M^2\right) = E\left(\Delta d_l^2\right) \cdot \frac{2M}{2 - \frac{I_{CP}K_d}{C_f}} \quad (16)$$

An approach similar to the one used in this Appendix can be used on any of the difference equation sets given in this paper.

REFERENCES

[1] G. Chien and P.R. Gray, "A 900-MHz Local Oscillator Using a DLL-Based Frequency Multiplier Technique for PCS Applications," *IEEE Journal of Solid State Circuits*, vol. 35, pp. 1996–1999, December 2000.

[2] D.J. Foley and M.P. Flynn, "CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator," *IEEE Journal of Solid State Circuits*, vol. 36, pp. 417–423, March 2001.

[3] B. Kim and T.C. Weigandt and P.R. Gray, "PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design," in *Proc. Int. Symp. On Circuits and Systems*, June 1994.

[4] B. Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design*, ch. Design of Monolithic Phase-Locked Loops and Clock Recovery Circuits — A Tutorial. IEEE Press, 1996.