Low-Jitter Clock Multiplication: A Comparison Between PLLs and DLLs

Remco C. H. van de Beek, *Student Member, IEEE*, Eric A. M. Klumperink, *Member, IEEE*, Cicero S. Vaucher, *Senior Member, IEEE*, and Bram Nauta, *Member, IEEE*

***Abstract*—This paper shows that, for a given power budget, a practical phase-locked loop (PLL)-based clock multiplier generates less jitter than a delay-locked loop (DLL) equivalent. This is due to the fact that the delay cells in a PLL ring-oscillator can consume more power per cell than their counterparts in the DLL. We can show that this effect is stronger than the notorious jitter accumulation effect that occurs in the voltage-controlled oscillator (VCO) of a PLL. First, an analysis of the stochastic-output jitter of the architectures, due to the most important noise sources, is presented. Then, another important source of jitter in a DLL-based clock multiplier is treated, namely the stochastic *mismatch* in the delay cells which compose the DLL voltage-controlled delay line (VCDL). An analysis is presented that relates the stochastic spread of the delay of the cells to the output jitter of the clock multiplier. A circuit design technique, called impedance level scaling, is then presented which allows the designer to optimize the noise and mismatch behavior of a circuit, independently from other specifications such as speed and linearity. Applying this technique on a delay cell design yields a direct tradeoff between noise induced jitter and power usage, and between stochastic mismatch induced jitter and power usage.**

***Index Terms*—Circuit modeling, delay-locked loops (DLLs), frequency conversion, jitter, phase-locked loops (PLLs).**

I. Introduction

An important building block in almost all digital and mixed signal integrated circuits (ICs) is the clock multiplier. Its function is to multiply an incoming reference clock frequency by a certain factor, usually because no crystals are available with a clock frequency as high as needed on-chip. Also, when parallel data is to be serialized using a multiplexer, clock multiplication is needed to time the outgoing bits. In these applications, the quality of the multiplied clock with respect to timing jitter is an important specification [1], [2].

Apart from the usual integer-N PLL implementation of the clock multiplier, where a voltage controlled oscillator (VCO) is locked to a clean reference clock, architectures based on a delay-locked loop (DLL) have been successfully used recently as clock multipliers [3]–[5]. In such an architecture, which is schematically shown in Fig. 1, a voltage controlled delay line (VCDL) is locked to a clean reference signal. The extra timing information needed to generate the high-frequency clock is obtained by using a VCDL that consists of several tuneable delay cells, in this way generating multiple phases of the low-frequency clock. These phases are combined into one high-frequency clock using a circuit that is referred to as "edge combiner."

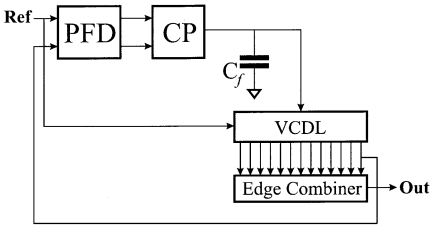

Fig. 1. DLL-based clock multiplier architecture.

As shown in the analysis presented in [6], the advantage of the DLL-based architecture is that the VCDL is "reset" with respect to stochastic jitter every time a new reference edge is applied at the input, whereas in the VCO of a PLL the jitter accumulates. This paper complements the analysis presented in [6] in several ways. First, by taking the effects of frequency multiplication into account. This paper examines structures where the output frequency is an integer multiple of the reference frequency. In this way, a PLL-based clock multiplier solution can be compared to a DLL-based clock multiplier, and new design considerations are obtained. Second, by including all important noise sources in the jitter analyses, opposed to the inclusion of only the VCDL-noise in the DLL and the VCO-noise in the PLL as done in [6].

This paper offers a set of design equations from which the output jitter can be predicted. This is done by first composing a mathematical model, based on difference equations, describing the behavior of the architectures. The output jitter due to different noise sources is then analyzed in the time domain directly.

Apart from jitter due to stochastic noise sources, which are examined first in this paper, the DLL-based architecture introduces a new source of timing errors, namely stochastic mismatch between the delay cells. This effect causes clock skew of the intermediate clock phases. The phenomenon will be measurable as systematic jitter on the high-frequency clock at the output of the edge combiner, and will appear as spurious signals in the output frequency spectrum of the clock multiplier [7]. The effects of delay cell mismatch on the jitter of the output signal are analyzed, resulting in a design equation for determining the feasibility of a DLL-based clock multiplier implementation. Because the mismatch parameters depend on the chip area of the

Manuscript received April 22, 2002; revised October 7, 2002. This paper was recommended by Associate Editor A. Hajamiri.

R. C. H. van de Beek, E. A. M. Klumperink, and B. Nauta are with the University of Twente, 7500 AE Enchede, The Netherlands (e-mail: r.c.h.vandebeek@utwente.nl).

C. S. Vaucher is with Philips Research Laboratories, Eindhoven 5656AA, The Netherlands.

Digital Object Identifier 10.1109/TCSII.2002.806248

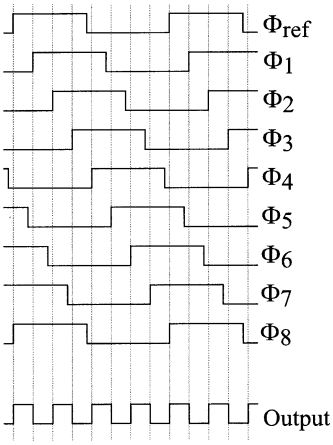

Fig. 2. Edge combination process for $N = 4$, using only rising edges to generate the output clock.

devices, the effect of scaling on the delay cell mismatch is then analyzed, using a technique called impedance level scaling [8]. This design technique proves useful in decoupling the noise and mismatch properties of a circuit from other properties such as speed or linearity.

Section II analyses the stochastic jitter of a DLL-based clock multiplier. Section III examines the stochastic jitter of an integer-N PLL. The structures are then compared in Section IV. In Section V, we analyze the effects of delay cell mismatch, and in Section VI, the impedance level scaling technique. In Section VII, simulation results verifying the analyses performed in this paper are discussed. The paper finally concludes in Section VIII with a summary of the results.

II. Analysis of DLL Jitter Due to Noise

In this section, the effect of different sources of stochastic DLL output jitter is analyzed. First, a mathematical model of the DLL is derived, which is then used to calculate the output jitter due to different noise sources in the architecture.

A. DLL Architecture

Fig. 1 shows the general architecture of a DLL with edge combiner. The feedback mechanism 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, making a loop filter zero unnecessary.

The basic idea behind a DLL-based clock multiplier is that the total delay of the multitapped VCDL 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 the input clock which 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 Fig. 2, where the frequency multiplication factor N equals four

If only the rising edges of the different clock phases are used to generate both the rising and falling edges of the generated

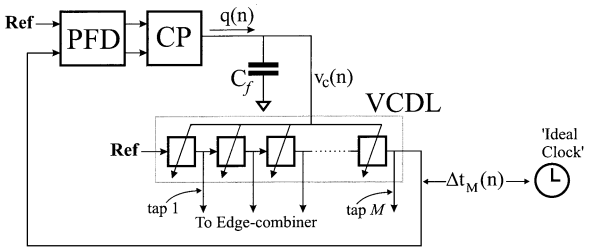

Fig. 3. DLL model that is used; the "ideal clock" illustrates the jitter definition used here.

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. However, timing dependency on the duty cycle of the reference is now introduced, which is a problem in some applications.

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 [3]. 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 (Fig. 2 being an example of this), and, thus, the number of delay cells M in the VCDL equals

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

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

B. 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 [9]. In this work, a very simple and intuitive definition will be used:

> Jitter is the random or systematic 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 Fig. 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 PFD compares the zero-crossing times of the reference to those of the last tap of the VCDL. The CP converts the measured time difference into a charge $q(n)$ which is pumped into the loop filter (a simple capacitor), thus, integrating this charge. Note, that parameter n indicates the period number of the input clock;

this variable is used in the difference equations that are derived shortly.

The DLL noise analysis depends on a number of assumptions which are listed below.

1) The loop has successfully locked to the state in which the VCDL delay equals the period time of the reference clock. This implies that the loop is stable.
2) The mean VCDL control voltage in lock equals 0 V. This simplifies analysis, while the results of the jitter calculations do not depend on this assumption because of the linearity of the system.
3) The current the CP delivers can be modeled by charge pulses with a dirac-pulse shape, which is allowed if the jitter is small compared to the reference period time.
4) All noise sources are white. This implies assuming no correlation between the noise contribution of a noise source in a certain period of the reference clock and previous contributions of the same source. A general statement about the validity of this assumption is hard to make. In theory, $1/f$ noise in the CP for example yields infinite jitter if integrated starting from DC. In practice however, there will be a lower limit on the frequency from which to integrate the phase noise, depending on measurement time or system specifications, bounding the jitter. Using conventional continuous modeling of the DLL behavior and a reasonable lower integration limit, it can be shown that the $1/f$ corner frequency should be one to two decades below the DLL bandwidth for the white noise to be dominant (for example, if the phase noise is to be integrated from 1 kHz up to 10 MHz and the $1/f$ corner frequency is at 1 MHz, the white noise energy is already dominant).
5) All noise sources are uncorrelated to the other noise sources in the loop.
6) The jitter contributed in a certain period of the input clock by a certain delay cell is not correlated to that delivered by another delay cell.
7) The variance of the jitter of every delay cell is equal. This is reasonable if all delay cells are realized equally and if the input signal shape of every delay cell is the same.
8) The loop behavior is linear, meaning that the output jitter contributions of every noise source can be calculated separately. The total jitter can then be calculated by adding the different contributions power-wise. This assumption is reasonable as long as the jitter remains low.

The tuning voltage v_c determines the delay of the VCDL $d_{\rm tot}$ according to

$$d_{\rm tot} = T_{\rm ref} - K_d v_c + \Delta d_{\rm tot} \tag{2}$$

where $T_{\rm ref}$ equals the period time of the clock, K_d is the gain of the VCDL, expressed in $[s \cdot {\rm V}^{-1}]$, and $\Delta d_{\rm tot}$ is the jitter added by the VCDL.

Deviations in the tuning voltage, as well as jitter added by the delay cells will result in jitter on the taps of the VCDL. Also, jitter present on the reference clock that is fed into the VCDL causes jitter on the output taps. Using the assumptions given before, 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 VCDL, which means the jitter variance will be highest at the last output tap.

The charge that is pumped into the loop filter capacitor by the CP is given by

$$q(n) = I_{\rm CP}\left\{\Delta t_M(n) - \Delta t_{\rm ref}(n) + \Delta t_{\rm PFD}(n)\right\} + q_{\rm noise}(n) \tag{3a}$$

where $q(n)$ is the charge that the CP pumps into the loop filter after input period number n with $q_{\rm noise}$ denoting the part of that charge caused by a noisy CP, $I_{\rm CP}$ is the CP current, $\Delta t_M(n)$ is the jitter at the last (Mth) output tap of the VCDL after the nth input period and $\Delta t_{\rm PFD}(n)$ is the detection error that the PFD makes due to its input referred voltage noise, which will be discussed in more detail later. The term $\Delta t_{\rm ref}(n)$ denotes the timing error in the reference edge that appears at the PFD input after input period number n.

Knowing the charge that is pumped into the filter, the VCDL control voltage during the nth input period is given by

$$v_c(n) = v_c(n-1) + \frac{q(n-1)}{C_f} \tag{3b}$$

with C_f the value of the loop filter capacitor.

The final difference equation describes the timing error of the last output tap Δt_M, using (2)

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

where $\Delta d_l(n)$ is the jitter added by the lth delay cell in input period number n. The reference jitter is visible at the last output tap after one clock period delay.

C. DLL Output Jitter Due to Noise

In this section the jitter that will result at the different output taps of the VCDL due to its own jitter is analyzed first, using the set of difference equations (3a)–(3c). Then, in a similar fashion, the output jitter due to the PFD and CP noise is calculated as well as the output jitter due to the reference jitter. The general calculation method is demonstrated in the Appendix.

To isolate the effect of the delay cell noise, the other noise sources are neglected, using Assumption 8 in Section II-B.

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

$$\sigma^2_{\Delta t_M} = \sigma^2_{\Delta d} \cdot M \frac{2}{2 - \varepsilon_{\rm DLL}} \tag{4}$$

with the so called *normalized-loop bandwidth* $\varepsilon_{\rm DLL}$ defined as [6]

$$\varepsilon_{\rm DLL} \equiv \frac{I_{\rm CP} K_d}{C_f} \approx \omega_0 T_{\rm ref}. \tag{5}$$

The approximation shows the relation between the value of $\varepsilon_{\rm DLL}$ and the DLL-loop bandwidth ω_0 [10].

Note, that (4) is in agreement with the result achieved in [6].

It is important to note that the jitter is lowest for low values of the DLL normalized loop bandwidth $\varepsilon_{\rm DLL}$, in which case the

jitter would be equal to that of a VCDL that is not controlled by a loop. This shows that the function of the control loop is not to remove jitter from the VCDL but merely to tune the total delay of the VCDL to the desired value.

Apart from the jitter that is generated by the VCDL, the loop components that take care of the feedback mechanism also introduce jitter. First, the PFD that has to detect zero-crossings is realized using noisy elements. The internal noise of the PFD 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, assuming that the incoming edges are not infinitely steep. Also, the 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. Both building blocks, thus, cause noise on the VCDL control voltage, resulting in output jitter.

To simplify calculations, the CP noise is calculated back to the input of the PFD as an equivalent time error

$$\Delta t'_{\mathrm{PFD}}(n) \equiv \Delta t_{\mathrm{PFD}}(n) + \frac{q_{\mathrm{noise}}(n)}{I_{\mathrm{CP}}}. \tag{6}$$

Using a method similar to the calculation of the jitter due to VCDL noise as described in the Appendix, the variance of the output jitter due to the PFD and CP noise can be calculated. This results in

$$\sigma^2_{\Delta t_M} = \sigma^2_{\Delta t'_{\mathrm{PFD}}} \frac{\varepsilon_{\mathrm{DLL}}}{2 - \varepsilon_{\mathrm{DLL}}}. \tag{7}$$

Applying the same method to analyze the jitter at the DLL output resulting from the jitter that is present in the reference signal at the input of the DLL yields

$$\sigma^2_{\Delta t_m} = \sigma^2_{\Delta t_{\mathrm{REF}}} \cdot \left\{ 1 + \frac{4\varepsilon_{\mathrm{DLL}}}{2 - \varepsilon_{\mathrm{DLL}}} \right\} \tag{8}$$

showing that a DLL can never decrease the jitter of the input reference, as is possible when using a PLL, because the jitter that is at the input of the VCDL will also be at the output of the taps. In fact, the deviations in the control voltage of the VCDL that are caused by the reference jitter will even increase the DLL output jitter.

From these equations it is again apparent that a small value of $\varepsilon_{\mathrm{DLL}}$ is beneficial for the DLL output jitter. The gain of the VCDL K_d should however be large enough to compensate for process spread and temperature variations; the CP current cannot be chosen too small because of the jitter resulting from mismatch in the CP. This means that the loop-filter capacitor should be made large at the cost of area, in order to maintain a reasonably low-loop bandwidth. Other practical issues such as settling behavior may also set a lower limit on the value of the loop bandwidth.

III. PLL Jitter Analysis

In this section, an analysis is presented, similar to that of DLL jitter, which applies to an integer-N PLL-based clock multiplier. The analysis starts by deriving difference equations describing the architecture, which are then used to calculate the PLL output jitter due to different noise sources in the time domain directly.

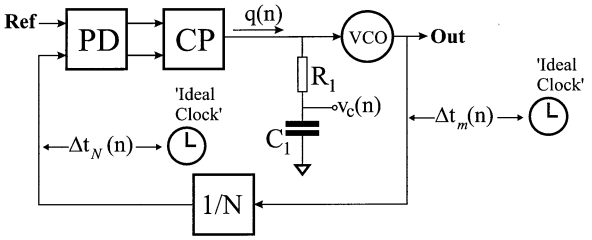

Fig. 4. PLL architecture.

A. Mathematical Model of the PLL With Noisy Building Blocks

The difference equations describing the behavior of the PLL mathematically are derived using the PLL model shown in Fig. 4. Again, the "ideal clocks" are merely shown to show the jitter concept used here.

The PLL noise analysis depends on a number of assumptions similar to those made for the DLL.

1) The PLL is in lock. This implies that the loop is stable.
2) The mean VCO control voltage in lock equals 0 V. This means that the free-running frequency of the VCO is exactly equal to N times the reference frequency. The results of the jitter calculations do not depend on this assumption because of the linearity of the system.
3) The current the CP delivers can be modeled by charge pulses with a Dirac-pulse shape.
4) All noise sources are white. From conventional PLL noise analysis one can conclude that this assumption is reasonable for a wide-band PLL. See also the remarks under Assumption 4 in Section II-B.
5) All noise sources are uncorrelated to other noise sources in the loop.
6) The loop behavior is linear, meaning that the superposition principle holds.

The variable n shown in the PLL model denotes the period number of the reference clock and is used in the difference equations that are to be derived.

As soon as the loop is in lock, the CP delivers current to the loop filter only just before and after a rising edge of the reference input signal, making the PLL behave much like a sampled system. To be able to model the behavior of this system, it is important to know the response of the loop filter and the VCO to a charge pulse from the CP. As stated in Assumption 3, this charge pulse is modeled by a Dirac current pulse, which is reasonable in most cases as the actual duration of this charge pulse is much shorter than one VCO period in practice [11], [12].

The angular frequency of the VCO is controlled by the VCO's control voltage v_{tune} such that

$$\omega_{\mathrm{VCO}}(t) = \omega_{\mathrm{fr}} + K_{\mathrm{VCO}} v_{\mathrm{tune}}(t) \tag{9}$$

where ω_{fr} is the free-running angular frequency of the VCO and K_{VCO} the VCO gain. The results of the jitter calculations do not depend on the value of the free-running frequency.

One can prove that the VCO output phase some time after a charge pulse (which occurs at $t = 0$) can be described mathematically as [12]

$$\phi_{\rm VCO}(t) \approx \omega_{\rm fr} t + K_{\rm VCO} R_1 \left(1 + \frac{t}{R_1 C_1}\right) q + K_{\rm VCO} v_c\left(0^-\right) t + \phi_{\rm VCO}\left(0^-\right) \quad (10)$$

with q the amount of charge pumped into the loop filter, $v_c(0^-)$ the loop filter capacitor voltage just before $t = 0$ and $\phi_{\rm VCO}(0^-)$ the VCO phase just before $t = 0$.

In practice, the zero-crossing time error of the VCO output can be estimated well by sampling the VCO phase at the *ideal* zero-crossing moments (which are the positive zero-crossing moments of a clock with a phase $\phi_{\rm ideal} = \omega_{\rm fr} t$). Using

$$\Delta t \approx -\frac{\phi_{\rm VCO} - \phi_{\rm ideal}}{2\pi} \frac{T_{\rm ref}}{N} \quad (11)$$

the zero-crossing time error of the mth positive zero-crossing of the VCO after the charge injection can be estimated well by

$$\Delta t_m \approx \Delta t\left(0^-\right) - \frac{K_{\rm VCO} R_1 T_{\rm ref}}{2\pi N}\left\{1 + \frac{m T_{\rm ref}}{N \cdot R_1 C_1}\right\} q - \frac{m K_{\rm VCO} T_{\rm ref}^2}{2\pi N^2} v_c\left(0^-\right). \quad (12)$$

In this equation, $\Delta t(0^-)$ is the VCO timing error just before the occurrence of the charge pulse.

The jitter variance will be highest for the edge that causes a rising edge at the output of the divider. This is because that edge is used by the loop to correct the VCO, so the timing error of the very next edge will be less. The edges following will again be more and more polluted by jitter as the loop is "dead" until the next comparison action.

Now a set of difference equations describing the loop behavior can be formulated

$$q(n) = I_{\rm CP} \cdot \Big\{\Delta t_N(n) + \Delta t_{\rm PFD}(n) + \Delta t_{\rm div}(n) - \Delta t_{\rm ref}(n)\Big\} + q_{\rm noise}(n) \quad (13a)$$

$$v_c(n) = v_c(n-1) + \frac{q(n-1)}{C_1} \quad (13b)$$

$$\Delta t_m(n) = \Delta t_N(n-1) - v_c(n)\frac{m K_{\rm VCO} T_{\rm ref}^2}{2\pi N^2} - \frac{K_{\rm VCO} R_1 T_{\rm ref}}{2\pi N} q(n-1) + \sum_{l=1}^{m} \Delta T_{{\rm VCO},l}(n). \quad (13c)$$

In these equations, $I_{\rm CP}$ denotes the CP current, the jitter introduced by the frequency divider is denoted by $\Delta t_{\rm div}(n)$, $\Delta t_{\rm ref}(n)$ is the deviation of the reference input compared to an ideal clock with a period time of $T_{\rm ref}$, $q_{\rm noise}(n)$ is the charge noise of the CP and $\Delta T_{{\rm VCO},l}(n)$ is the period error of the VCO in its lth cycle within a reference period, both due to internal noise of the VCO and the voltage noise on the control line of the VCO generated by the resistor of the loop filter.

B. PLL Output Jitter Due to Noise

In this section, the jitter caused by random VCO period variations is analyzed first using the set of difference equations given by (13a)–(13c). All other sources of jitter are assumed to be zero in this analysis. The effect of the other sources of jitter in a PLL are then discussed briefly.

Using a procedure similar to the example in the Appendix leads to the following value of the jitter variance of the PLL output signal due to VCO jitter:

$$\sigma_{\Delta t_N}^2 = E\left(\Delta t_N^2\right) = \sigma_{\Delta T_{\rm VCO}}^2 \frac{N}{\varepsilon_{\rm PLL}\left\{2 - \varepsilon_{\rm PLL}\left(1 + \frac{T_{\rm ref}}{2R_1C_1}\right)\right\}} \quad (14)$$

where $\sigma_{\Delta T_{\rm VCO}}^2$ symbolizes the variance of the VCO period jitter, as it would occur for a free-running VCO.

Again, $\varepsilon_{\rm PLL}$ denotes the *normalized loop bandwidth*. This quantity is a design variable that is defined in the case of a PLL as [6]

$$\varepsilon_{\rm PLL} \equiv \frac{I_{\rm CP} K_{\rm VCO} R_1 T_{\rm ref}}{2\pi N} \approx \omega_0 T_{\rm ref} \quad (15)$$

where ω_0 now denotes the PLL bandwidth. Note, that this definition is different from the one used for the DLL; in both cases, however, ε denotes the normalized loop bandwidth of the structure.

In practical PLL designs, the position of the loop filter zero is much smaller than the reference frequency. This means that $T_{\rm ref}/(2R_1C_1)$ can be considered to be negligible to one, reducing (14) to

$$\sigma_{\Delta t_N}^2 = E\left(\Delta t_N^2\right) = \sigma_{\Delta T_{\rm VCO}}^2 \frac{N}{\varepsilon_{\rm PLL}\left(2 - \varepsilon_{\rm PLL}\right)} \quad (16)$$

which agrees with [6], where the same assumption was used.

It is interesting to see that for $\varepsilon_{\rm PLL} < 1$ the maximum output jitter of a PLL is smaller with a large normalized loop bandwidth $\varepsilon_{\rm PLL}$ (provided that the jitter is most dominantly due to internal VCO noise). This observation corresponds with the well-known fact that VCO noise can be cleaned up with a wide-band PLL.

Note, that the VCO noise is not the only source of output jitter. The internal noise of the building blocks other than the VCO will cause variations on the VCO tuning voltage, and, thus, output jitter. To ease calculations, the noise of the other PLL building blocks is calculated back to the input of the PFD according to

$$\Delta t_{\rm synth}(n) \equiv \Delta t_{\rm PFD}(n) + \Delta t_{\rm div}(n) + \frac{q_{\rm noise}(n)}{I_{CP}} \quad (17)$$

the variance of which is referred to as $\sigma_{\Delta t_{\rm synth}}^2$.

The PLL jitter due to these noise sources can now be shown to be (setting the other noise sources to zero)

$$\sigma_{\Delta t_N}^2 = \sigma_{\Delta t_{\rm synth}}^2 \frac{\varepsilon_{\rm PLL} + \frac{T_{\rm ref}}{2R_1C_1}\left(2 + \varepsilon_{\rm PLL}\right)}{2 - \varepsilon_{\rm PLL}\left(1 + \frac{T_{\rm ref}}{2R_1C_1}\right)}. \quad (18)$$

Very similarly, the jitter on the reference signal will cause jitter on the PLL output signal, according to

$$\sigma^2_{\Delta t_N} = \sigma^2_{\Delta t_{\rm ref}} \frac{\varepsilon_{\rm PLL} + \frac{T_{\rm ref}}{2R_1C_1}(2+\varepsilon_{\rm PLL})}{2-\varepsilon_{\rm PLL}\left(1+\frac{T_{\rm ref}}{2R_1C_1}\right)}. \quad (19)$$

Observing these equations leads to the conclusion that, contrary to the VCO induced jitter, a large value of $\varepsilon_{\rm PLL}$ (corresponding to a large PLL bandwidth) will raise the PLL output jitter due to the noise of the other loop components.

Finally, the loop-filter resistor will cause thermal noise at the input of the VCO, which is measurable at the PLL output as jitter. Using the fact that the thermal noise of the resistor is integrated by the VCO during every VCO period (which lasts approximately $T_{\rm ref}/N$), the variance of the VCO period deviation caused by this thermal noise can be shown to be

$$\sigma^2_{\Delta T_{\rm VCO}} = kTR_1 \frac{K^2_{\rm VCO}T^3_{\rm ref}}{2\pi^2N^3} \quad (20)$$

where k is the Boltzmann constant and T the absolute temperature.

Substituting this in (16) yields

$$\sigma^2_{\Delta t_N} = kT\frac{K_{\rm VCO}T^2_{\rm ref}}{\pi N\cdot I_{\rm CP}}\frac{1}{(2-\varepsilon_{\rm PLL})} \approx kT\frac{K_{\rm VCO}T^2_{ref}}{2\pi N\cdot I_{CP}} \quad (21)$$

where the last approximation holds for small values of the normalized PLL loop bandwidth.

C. PLL Optimization

As was shown before, a larger value of $\varepsilon_{\rm PLL}$ will *lower* the output jitter due to VCO phase noise while *raising* the jitter contribution of the other synthesizer noise sources. It is, thus, to be expected that there will be an optimum value for $\varepsilon_{\rm PLL}$. To be able to compare the DLL jitter characteristics with those of the PLL, the PLL should first be optimized.

To simplify things, we assume that $\varepsilon_{\rm PLL}$ is much smaller than two and that $T_{\rm ref}/(2R_1C_1)$ is negligible to one. Then the total PLL output jitter can be approximated by

$$\sigma^2_{\Delta t_N} = \sigma^2_{\Delta T_{\rm VCO}}\frac{N}{2\varepsilon_{\rm PLL}} + \sigma^2_{\Delta t_{\rm synth}}\frac{\varepsilon_{\rm PLL}}{2}. \quad (22)$$

The smallest amount of jitter is found for

$$\varepsilon_{\rm PLL,opt} = \frac{\sqrt{N\cdot\sigma^2_{\Delta T_{\rm VCO}}}}{\sqrt{\sigma^2_{\Delta t_{\rm synth}}}} \quad (23)$$

for which the total jitter can be approximated by

$$\sigma^2_{\Delta t_N} = \sqrt{N\cdot\sigma^2_{\Delta T_{\rm VCO}}\cdot\sigma^2_{\Delta t_{\rm synth}}}. \quad (24)$$

It is important to note that if the PLL bandwidth equals $\varepsilon_{\rm PLL,opt}$ the jitter due to the VCO equals the jitter that is caused by the other loop components.

IV. Comparison Between DLL and PLL Stochastic Jitter

In practical PLL-based clock multipliers, the VCO is often realized by a ring-oscillator as opposed to an oscillator using an LC-tank for frequency stability. An important reason for this is the area consumption of the on-chip inductor, but also portability to newer processes and oscillator pulling effects are arguments against an LC-oscillator. An important disadvantage of a ring-oscillator is the relatively high jitter it produces, which is to be cleaned up by using a wide-band PLL [13], [14]. The maximum bandwidth of a PLL is in practice limited by stability considerations to about one tenth of the reference frequency that is used at the input of the PFD [11], [13]. Expressed in terms of the normalized loop bandwidth $\varepsilon_{\rm PLL}$, this leads to

$$\varepsilon_{\rm PLL,max} \approx \omega_{0_{\rm max}}T_{\rm ref} \approx \frac{2\pi f_{\rm ref}}{10}T_{\rm ref} = \frac{1}{5}\pi \approx 0.63. \quad (25)$$

Because of better supply noise and substrate bounce rejection, differential delay cells are often used in the ring-oscillator of the PLL. To compare the output jitter of an integer-N PLL to the DLL-based architecture, we assume that both the VCDL and the VCO consist of delay cells of similar topology: each delay cell consists of an NMOS differential pair with resistive load.

The jitter of the ring oscillator can be predicted using the analysis presented in [15]. An important result from this work is

$$\frac{\sigma_{\Delta t_d}}{t_d} = \sqrt{\frac{kT}{C_L}}\frac{\xi}{V_{\rm GS}-V_T} \quad (26)$$

in which $\sigma_{\Delta t_d}$ is the rms-jitter of the cell, t_d is the delay of the cell, C_L is the load capacitance of one delay cell, ξ is a factor determined by the design, and $V_{\rm GS}-V_T$ the overdrive voltage of the NMOS differential pair transistors.

Knowing that the delay of one cell can be written as [15]

$$t_d = V_{\rm pp}\frac{C_L}{I_{\rm SS}} \quad (27)$$

with $V_{\rm pp}$ the peak-to-peak voltage swing of the delay cell and $I_{\rm SS}$ the static current it consumes, we can rewrite (26) as

$$\sigma^2_{\Delta t_d} = \left\{\frac{\xi^2kTV_{\rm pp}}{(V_{\rm GS}-V_T)^2}\right\}\frac{t_d}{I_{\rm SS}} = c\frac{t_d}{I_{\rm SS}} \quad (28)$$

with c a design dependent constant with unit $[A\cdot s]$ representing the bracketed part.

Using this equation, it is easy to show that the period jitter of a ring oscillator constructed using these delay cells is

$$\begin{aligned}\sigma^2_{\Delta T_{\rm VCO}} &= cV_{DD}M_{\rm VCO}\frac{T_{\rm VCO}}{P_{\rm static,VCO}}\\ &= cV_{\rm DD}\frac{M_{\rm VCO}}{N}\frac{T_{\rm ref}}{P_{\rm static,VCO}}\end{aligned} \quad (29)$$

where $V_{\rm DD}$ is the supply voltage of the oscillator, $M_{\rm VCO}$ the number of delay cells used in the VCO, $T_{\rm VCO}$ the period time of the oscillator and $P_{\rm static,VCO}$ the static power used in the VCO.

For simplicity, we first assume that the VCO is the most dominant source of jitter in the PLL (the other jitter sources will be included later in the comparison for completeness). Then, using (16) we can write

$$\sigma^2_{\Delta t_N} = c \cdot M_{\mathrm{VCO}} T_{\mathrm{ref}} \frac{V_{\mathrm{DD}}}{P_{\mathrm{static,VCO}}} \frac{N}{\varepsilon_{\mathrm{PLL}}\left(2-\varepsilon_{\mathrm{PLL}}\right)}. \quad (30)$$

The jitter of a DLL used to multiply the reference by the same factor N can be estimated by (4), which reduces to

$$\sigma^2_{\Delta t_M} = \sigma^2_{\Delta d_l} \cdot M_{\mathrm{VCDL}} \quad (31)$$

for small values of the normalized DLL-loop bandwidth.

Again, the jitter per delay cell can be predicted using (28), yielding

$$\sigma^2_{\Delta t_M} = cM_{\mathrm{VCDL}} \frac{t_d}{I_{\mathrm{SS}}} = 2cNT_{\mathrm{ref}} \frac{V_{\mathrm{DD}}}{P_{\mathrm{static,VCDL}}}. \quad (32)$$

Now if we allow an equal power usage in both the VCO and the VCDL, comparing (30) to (32) yields

$$N > \frac{M_{\mathrm{VCO}}}{2\varepsilon_{\mathrm{PLL}}\left(2-\varepsilon_{\mathrm{PLL}}\right)} \Rightarrow \left(\sigma^2_{\Delta t}\right)_{\mathrm{DLL}} > \left(\sigma^2_{\Delta t}\right)_{\mathrm{PLL}}. \quad (33)$$

If we assume a VCO consisting of three delay cells and a PLL with a normalized loop bandwidth given by (25), this leads to the conclusion that if the frequency multiplication factor N is higher than about 1.74, the DLL output jitter will be higher than the PLL output jitter. Because N is in practice an integer number, we can draw the conclusion that under the assumptions given in this section a PLL-based clock multiplier yields less output jitter than a DLL-based clock multiplier. This is because spending the same amount of power in the VCO as in the VCDL yields more power in the VCO *per delay cell* and, thus, less jitter per cell. This effect is larger than the jitter accumulation factor discussed in [6] (and expressed in (16) by the term $1/\{\varepsilon_{\mathrm{PLL}}(2-\varepsilon_{\mathrm{PLL}})\}$), which is not much larger than one for a wide-band PLL.

It is possible to get rid of the jitter accumulation in a PLL by periodically aligning the VCO with the reference signal, as shown in [16], [17]. This makes the loop behave more like a DLL in which the delay cells are reused within one cycle of the reference clock, enabling more power usage *per cell*. This frequency multiplication technique does not need an edge combiner to increase the frequency. A disadvantage of this principle is that the injection of the reference clock should be timed very accurately, which might require calibration. This required timing accuracy might make the technique unsuitable for very high-frequency clocks.

For completeness, an equation is derived that is valid for a PLL with additional jitter sources. The simplest way of doing this, is to realize that if the PLL bandwidth has been optimized with respect to jitter, the total output jitter is twice the jitter due to the VCO, as noted before. If we again assume that the VCO power consumption equals that of the VCDL of the DLL, (33) can be rewritten as

$$N > \frac{M_{\mathrm{VCO}}}{\varepsilon_{\mathrm{PLL,opt}}\left(2-\varepsilon_{\mathrm{PLL,opt}}\right)} \Rightarrow \left(\sigma^2_{\Delta t}\right)_{\mathrm{DLL}} > \left(\sigma^2_{\Delta t}\right)_{\mathrm{PLL}}. \quad (34)$$

Using the results of the PLL optimization in this equation leads to the following conclusion:

$$N > \frac{1}{2}\sqrt{\frac{M_{\mathrm{VCO}}P_{\mathrm{static}}}{cT_{\mathrm{ref}}V_{\mathrm{DD}}}\sigma^2_{\Delta t_{\mathrm{synth}}}} \Rightarrow \left(\sigma^2_{\Delta t}\right)_{\mathrm{DLL}} > \left(\sigma^2_{\Delta t}\right)_{\mathrm{PLL}} \quad (35)$$

where both the power used by the VCO and by the VCDL are equal to P_{static}. We can conclude that the more dominant the noise sources other than the VCO are in the PLL, the higher the frequency multiplication factor N is that is needed for the PLL to be superior to the DLL with respect to stochastic noise. Note, that reference jitter is not included in this equation; the PLL is always superior to the DLL with respect to jitter transfer.

We have assumed that the dominant power usage of the delay cells is static and that the jitter of the cells is mostly due to thermal noise. For practical implementations, these assumptions are often reasonable. However, if the delay cells consist of for example CMOS inverters, where power usage does not depend on delay line length and consequently not on the frequency multiplication factor, the DLL will perform somewhat better than the PLL, due to jitter accumulation. This also holds when the jitter is mostly caused by supply or substrate noise [9] as the jitter cannot be lowered by raising the power then. In both cases, the difference is small however, as the accumulation factor of a wide-band PLL is not much larger than one.

V. DLL Output Jitter Due to Delay Cell Mismatch

Because of stochastic component mismatch, the delay of different delay cells in the VCDL of a DLL will not be exactly equal for a certain tuning voltage, which will result in jitter as all the intermediate edges on the different output taps are not corrected by the loop. The amount of jitter caused by this effect is calculated here.

Although mismatch is caused by a stochastic process, the jitter that originates from it is deterministic, because once the chip has been processed, the mismatch properties are more or less fixed. Knowing the stochastic properties of the mismatch, predictions can be made *a priori* about the deterministic jitter.

The delay mismatch can be described mathematically as follows:

$$d_i = \{1 + e_i(v_c)\}\, d_{\mathrm{tune}} \quad (36)$$

where d_i is the particular delay of delay cell number i, d_{tune} is some nominal delay which is controlled by the VCDL tuning voltage v_c and $e_i(v_c)$ is a random variable, describing the delay cell mismatch for a certain value of v_c. For simplicity, this dependency on v_c will not be shown explicitly in the remaining equations. The variable e_i is assumed to have zero mean. This is reasonable as any common change of delay in the cells is removed by the loop. The delay mismatch of different cells is assumed to be uncorrelated.

The total delay of the VCDL will be equal to one period of the input clock after lock has been achieved. This results in the following equation for the individual delay of the delay cells:

$$d_i = T_{\mathrm{ref}} \frac{1+e_i}{M + \sum_{i=1}^{M} e_i} \quad (37)$$

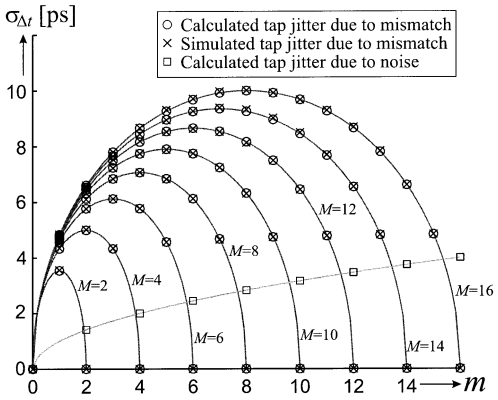

Fig. 5. Numerical statistical simulation results of the DLL jitter due to delay cell mismatch.

where M denotes the number of delay cells in the VCDL and $T_{\rm ref}$ the period time of the reference signal.

Now an expression for the total systematic jitter of the signal on the mth tap (at the output of the mth delay cell) can be derived. If all the delay cells would be perfectly matched, the delay between the input and the mth tap would be $(m/M)T_{\rm ref}$. In case of mismatch, the systematic jitter after m cells can then be calculated to be

$$\Delta t_m = \sum_{i=1}^{m} d_i - \frac{m}{M}T_{\rm ref} = T_{\rm ref}\left(\frac{m+\sum_{i=1}^{m} e_i}{M+\sum_{i=1}^{M} e_i} - \frac{m}{M}\right) \quad (38)$$

the variance of which can be shown to be

$$\sigma^2_{\Delta t_m} = E\left\{(\Delta t_m)^2\right\} \approx T^2_{\rm ref}\frac{m(M-m)}{M^3}\sigma^2_{e_i} \quad (39)$$

assuming uncorrelated values of e_i with zero mean. A first-order Taylor expansion has been used, assuming $\sigma^2_{e_i} \ll 1$.

It is interesting to note that the variance of Δt_m is highest for $m = M/2$, i.e., halfway the VCDL. This is to be expected: the loop controls the VCDL such that the time error at its output is zero, while the error at the input of the VCDL is also zero. The highest timing uncertainty will be in the middle of the VCDL, where the distance to these clean points is highest. This is comparable to mismatch in resistors in a resistor string based A/D converter, where the highest deviation is also found in the middle of the string [18].

The sigma value of the phase time error halfway the VCDL can be approximated, using (39), to be

$$\sigma_{\Delta t_{M/2}} \approx \sigma_{e_i}\cdot\frac{T_{\rm ref}}{2\sqrt{M}}. \quad (40)$$

Equation (39) has been verified using numerical statistical analysis for a constant value of the nominal delay of a single delay cell, the results of which are shown in Fig. 5. This figure shows a very good agreement between the predicted time deviations and the simulations. It also clearly shows the peak of the time deviation variance at the middle of the VCDL.

The jitter due to delay cell noise is also shown in the figure, for an arbitrary value of $\sigma_{\Delta t_d}$, the rms-jitter of a single delay cell due to noise. Using the fact that DLL output jitter due to delay cell noise is approximately equal to the stochastic jitter of the *uncontrolled* VCDL [6] yields

$$\sigma_{\Delta t_m} \approx \sqrt{m}\sigma_{\Delta t_d} \quad (41)$$

showing that the effect of delay cell noise is highest on the *last* output tap, as opposed to mismatch induced jitter.

If we define a measure of relative jitter, where the sigma value of the maximum time deviation is related to the output period of the clock multiplier, the following result is obtained

$$\frac{\sigma_{\Delta t_{M/2}}}{\left(\frac{T_{\rm ref}}{N}\right)} \approx \sigma_{e_i}\cdot\frac{\sqrt{N}}{2\sqrt{2}} \quad (42)$$

using (1), which shows that the relative jitter of the output signal is proportional to the square root of the frequency multiplication factor N. This dependency on N was also shown for rms-jitter due to delay cell noise.

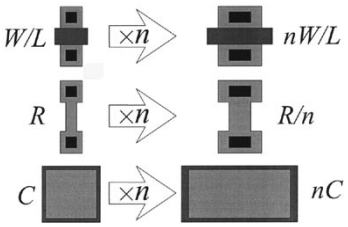

Fig. 6. Concept of impedance level scaling.

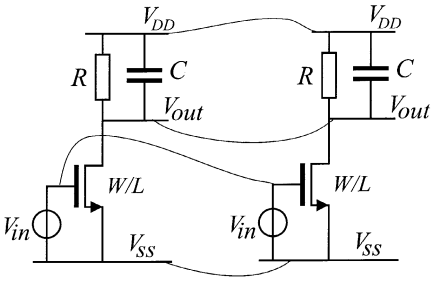

Fig. 7. Impedance level scaling presented as putting identical circuits in parallel.

VI. Impedance Level Scaling

It is a well-known fact that increasing the area of on-chip MOS-transistors improves the matching properties of those transistors [19]. The same also goes for the matching of resistors and capacitors on an IC [20]. This leads us to investigate the effect of increasing the area of a complete circuit in a systematic manner that we call *impedance level scaling*.

The concept of impedance level scaling is fairly simple, yet leads to very useful design considerations. This technique enables a decoupled optimization of the noise and mismatch properties of a circuit independent of other properties such as speed and linearity, thus, simplifying the task of the designer.

Starting from a circuit that has been optimized with respect to specifications other than noise and mismatch, one can scale the *width* of every component of that circuit by a certain factor α. This is shown conceptually in Fig. 6, where the effect on the component values is also shown.

Using the analogy that scaling is similar to putting identical circuits in parallel, as illustrated in Fig. 7 where $\alpha = 2$, it is

TABLE I
EFFECT OF IMPEDANCE LEVEL SCALING ON COMPONENT PROPERTIES

Starting circuit	After Impedance Level Scaling
v_n, W/L: $\sigma_{v_n} = \sqrt{\frac{4kT\gamma}{g_m}\Delta f}$, $\sigma_{\Delta\beta/\beta} = \frac{A_\beta}{\sqrt{W \cdot L}}$, $\sigma_{\Delta V_T} = \frac{A_{V_T}}{\sqrt{W \cdot L}}$	$g'_m = g_m \cdot \alpha$; v'_n, nW/L: $\sigma'_{v_n} = \sigma_{v_n} \cdot \frac{1}{\sqrt{\alpha}}$, $\sigma'_{\Delta\beta/\beta} = \sigma_{\Delta\beta/\beta} \cdot \frac{1}{\sqrt{\alpha}}$, $\sigma'_{\Delta V_T} = \sigma_{\Delta V_T} \cdot \frac{1}{\sqrt{\alpha}}$
v_n, R: $\sigma_{v_n} = \sqrt{4kTR\Delta f}$, $\sigma_{\Delta R/R} = \frac{A_R}{\sqrt{\text{Area}}}$	v'_n, nR: $\sigma'_{v_n} = \sigma_{v_n} \cdot \frac{1}{\sqrt{\alpha}}$, $\sigma'_{\Delta R/R} = \sigma_{\Delta R/R} \cdot \frac{1}{\sqrt{\alpha}}$
C: $\sigma_{\Delta C/C} = \frac{A_C}{\sqrt{\text{Area}}}$	nC: $\sigma'_{\Delta C/C} = \sigma_{\Delta C/C} \cdot \frac{1}{\sqrt{n}}$

easy to deduce that the node voltages of the scaled circuit are equal to those of the original circuit, provided the circuit is not heavily loaded externally. From this analogy it is also clear that the scaling will not change linearity and speed of the circuit.

A fact that is familiar to many designers is that impedance level scaling will improve the signal to noise ratio of the circuit at the cost of increased power usage. More precisely, scaling the circuit by a factor α will decrease the rms-value of the noise voltages by a factor $\sqrt{\alpha}$ while increasing the power usage by a factor α, meaning there is a direct tradeoff between power usage and noise.

A less familiar but important property of impedance level scaling is the effect it has on the mismatch errors of a circuit. Assume the relative change in the value of a certain component changes some circuit parameter (for example, the offset voltage, or the delay of a delay cell) *linearly*. This is reasonable as long as mismatch changes the value of a component just slightly. The same *relative* change of the corresponding component in the scaled circuit will result in the same change of the output parameter, which can again be understood by the scaling analogy depicted in Fig. 7. But the mismatch of the component value of the scaled circuit will reduce by a factor $\sqrt{\alpha}$ (see Table I), which means the sensitivity of circuit parameters such as offset and delay errors will be $\sqrt{\alpha}$ times less in the scaled circuit than in the starting circuit, at the cost of increased power usage.

For a delay cell, the implication of the impedance level scaling is that increasing the power by a factor α yields a stochastic jitter reduction of $\sqrt{\alpha}$ (which also follows from the jitter analysis in [15]). Also the mismatch of the delay between different cells will improve by a factor $\sqrt{\alpha}$.

VII. SIMULATION RESULTS

In this section, results of high-level DLL and PLL simulations are presented first. These simulations were performed to verify the equations that were derived for the output jitter due to stochastic noise sources. Then, results of Monte Carlo simulations of a delay line are shown. These were done to verify the predictions done about impedance level scaling and to give an indication of the severity of the mismatch induced jitter compared to jitter due to thermal noise.

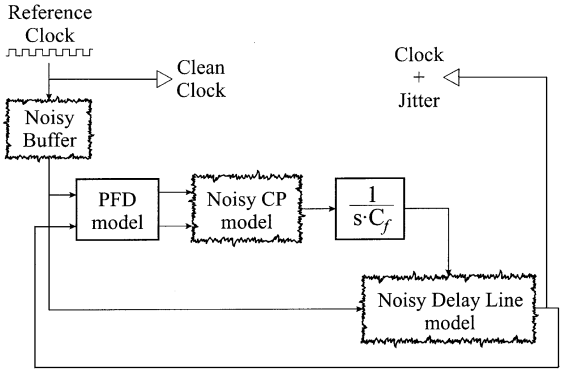

Fig. 8. Simulation model for the DLL.

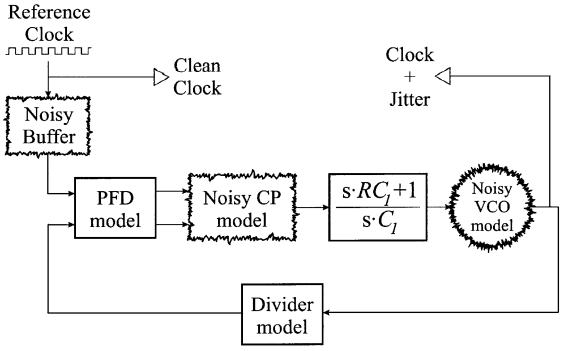

Fig. 9. Simulation model for the PLL.

A. Stochastic Jitter Simulations

To verify the stochastic jitter predictions that are described in the previous section, high-level simulation models of a DLL and a PLL have been used in Simulink (which is a MATLAB[1] simulation shell). These models are depicted in Figs. 8 and 9. Although these simulations were time consuming, enhancement of simulation speed using techniques such as described in [21] were not used, as these techniques do not apply to systems with additive noise.

The most important noise sources used in the analyses can be applied independently. The delay cell noise is modeled by random uncorrelated delay variations with zero mean. The CP noise is modeled by adding white noise to the CP 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 [22]). 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 and PLL output signals. The jitter is then calculated as the variance of the time differences.

[1]MATLAB is a registered trademark of The MathWorks, Natick, MA.

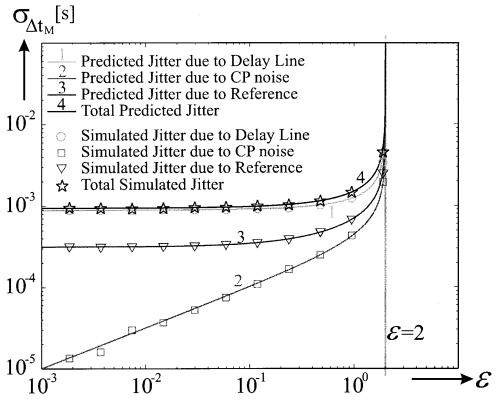

Fig. 10. DLL simulation results: output jitter versus loop bandwidth normalized to reference frequency.

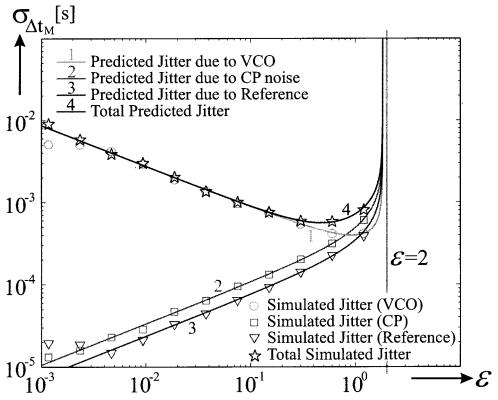

Fig. 11. PLL simulation results: output jitter versus loop bandwidth normalized to reference frequency.

The graphs shown in Figs. 10 and 11 show simulation results for a clock multiplication factor N of 8, meaning that the VCDL consists of 16 delay cells. The VCO consists of three delay cells. The jitter of the VCO delay cells was related to that of the VCDL delay cells according to (29) and (32).

First, simulations were done with only one noise source turned on with the variances of the other sources put to zero. The graphs show good agreement between the predicted and the simulated points. Then, all noise sources were turned on simultaneously to prove that the superposition principle, that was used as an important assumption throughout the analysis, was valid (meaning the jitter contribution of the different noise sources could be added power-wise). The result of these simulations is also shown, again showing good agreement with expectations. The deviations at low-normalized bandwidths are caused by the fact that the simulation time was short compared to the settling time at those bandwidths.

The simulation results give confidence in predictions of DLL and PLL output jitter based on the equations derived in this paper. They confirm the prediction that the PLL would have lower output jitter in its optimum than the DLL clock multiplier (in this case the total optimized PLL rms-jitter is roughly half that of the DLL).

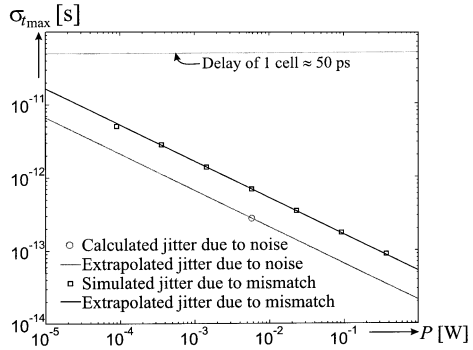

Fig. 12. Relation between power per delay cell and DLL jitter, due to noise and mismatch; $M = 16$, $T_{\rm ref} = 800$ ps.

B. Mismatch Simulations

Monte Carlo simulations have been performed in a SPICE-like simulation tool on a delay line in order to verify the effect of impedance level scaling on the delay mismatch and to compare the jitter due to mismatch to the jitter caused by circuit noise. The delay cells were realized as differential NMOS pairs with a resistive load, in a modern 0.18-μm CMOS process. The delay of a single cell was about 50 ps; the differential voltage swing was 500 mV. The delay cell mismatch spread was simulated for various values of the scale factor α. The results of the simulations are presented in Fig. 12, where the results are used in combination with (40) with $M = 16$ and $T_{\rm ref} = 800$ ps. The upper solid line through these points has been calculated by applying the scaling theory on the simulation point at $P =$ 5.8 mW. The graph shows good agreement between theory and simulations.

Using results presented in [15], it is possible to estimate the jitter of one delay cell due to circuit noise. This has been done using operation point information obtained from simulations of the cells at $P = 5.8$ mW. Using (41) leads to

$$\sigma_{\Delta t_M} \approx \sqrt{M}\sigma_{\Delta t_d} \tag{43}$$

where $\sigma_{\Delta t_d}$ is the rms jitter of a single delay cell as calculated in [15]. The calculated jitter due to noise is shown in Fig. 12, where the solid line represents the extrapolation of this calculation according to the scaling theory.

It is obvious from the graph that jitter due to mismatch is in this case dominating the jitter behavior of the delay line. Another important observation is that increasing the power has the same effect on both the jitter due to noise and the jitter due to mismatch (increasing the power per delay cell with a factor α, decreases the jitter by a factor of $\sqrt{\alpha}$). Because higher power usage leads to lower total jitter, it is in theory possible to meet strict jitter specifications with a DLL-based architecture. This might however lead to unrealistic power usage of the structure.

VIII. Conclusion

Although a DLL-based clock multiplier at first glance seems a better choice than a PLL based architecture because of the jitter accumulation effects in the PLL, the fact that the structures should perform clock multiplication leads to a drastically different conclusion. In practical implementations of clock multi-

pliers (based on either a DLL architecture or an integer-N PLL), the fact that the VCDL of the DLL needs more delay sections to perform the same task yields a lower power budget *per delay cell* for the VCDL than for the VCO and, thus, less jitter per delay cell. This effect is stronger than the jitter accumulation that the VCO of a PLL suffers from, leading to the conclusion that a wide-band PLL used for clock multiplication produces less output jitter than a DLL-based implementation of the same function. This conclusion is based on a wide-band PLL that uses a differential ring oscillator built using delay elements similar to those used in the VCDL of the DLL.

Another very important source of jitter should be taken into consideration for the DLL-based architecture: the stochastic mismatch of the delay cells in the VCDL. Monte Carlo simulations with a modern CMOS process indicate that this type of jitter is dominant in a DLL where intermediate clock phases of the VCDL are also used, due to the clock skew that is caused by the mismatch.

It has been shown, using the concept of impedance level scaling, that there is a direct tradeoff between power usage and output jitter of the frequency multiplier, both due to thermal noise and to mismatch. The amount of output jitter is limited directly by the power budget of the circuit. It can be shown that if the delay cell mismatch is the most dominant jitter source for a certain circuit, it will still be dominant in an impedance level scaled version of this circuit.

Finally, the analysis of the DLL has shown an important design consideration for this type of clock multipliers. The output jitter can be minimized by minimizing the DLL-loop bandwidth, showing 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 VCDL to be equal to the reference period. For a very small loop bandwidth, the DLL behaves as if uncontrolled with respect to jitter. For an integer-N PLL, the normalized PLL-loop bandwidth shows a certain optimum.

APPENDIX

To demonstrate how to obtain the output jitter of a system described by difference equations, the calculation of the output jitter of a DLL with a VCDL that consists of noisy delay cells is shown in this appendix. This is done using the set of difference equations given by (3a)–(3c) describing the DLL behavior mathematically. For this analysis, we use the assumptions given in Section II of this paper.

First, we assume that the noisy delay cells are the only source of jitter. The set of difference equations can then be reduced to

$$v_c(n) = v_c(n-1) + \frac{I_{CP}}{C_f}\Delta t_M(n-1) \tag{44a}$$

$$\Delta t_M(n) = -K_d v_c(n) + \sum_{l=1}^{M} \Delta d_l(n) \tag{44b}$$

substituting (3a) in (3b).

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

$$\begin{aligned}\sigma^2_{\Delta t_M} &= E\left\{\left(-K_d v_c(n) + \sum_{l=1}^{M}\Delta d_l(n)\right)^2\right\} \\ &= K_d^2 \cdot E\left(v_c^2(n)\right) - 2K_d \cdot E\left(v_c(n)\sum_{l=1}^{M}\Delta d_l(n)\right) \\ &\quad + E\left\{\left(\sum_{l=1}^{M}\Delta d_l(n)\right)^2\right\}. \end{aligned}\tag{45}$$

Because the variance of the tuning voltage does not depend on the period number n in the locked situation (in this situation, the output jitter is the result of a stationary process), this equation can now be reduced to

$$\sigma^2_{\Delta t_M} = E\left(\Delta t_M^2\right) = K_d^2 \cdot E\left(v_c^2\right) + M \cdot E\left(\Delta d^2\right) \tag{46a}$$

taking into account the variables in (45) that are uncorrelated. We also assume that the jitter of every delay cell has the same statistical properties, meaning that $E(\Delta d_l^2)$ does not depend on l and can be written as $E(\Delta d^2)$.

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 (44a). The following equation can be derived from it by taking the square on both the left- and right-hand side, followed by equating the expected value of both sides, taking into account the uncorrelated variables

$$E\left(v_c^2\right) = E\left(v_c^2\right) + 2\frac{I_{\mathrm{CP}}}{C_f}E\left\{v_c(n)\Delta t_M(n)\right\} + \frac{I_{\mathrm{CP}}^2}{C_f^2}E\left(\Delta t_M^2\right). \tag{46b}$$

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 (44b). Squaring this equation and equating the expected value of both the left- and right-hand side results in the needed new independent equation, making it possible to solve for the tuning voltage variance

$$\begin{aligned}K_d^2 \cdot E\left(v_c^2\right) + 2K_d \cdot E\left\{v_c(n)\Delta t_M(n)\right\} + E\left(\Delta t_M^2\right) \\ = M \cdot E\left(\Delta d^2\right)\end{aligned}\tag{46c}$$

Finally, solving the set of (46) for $E(\Delta t_M^2)$ results in

$$E\left(\Delta t_M^2\right) = E\left(\Delta d^2\right) \cdot \frac{2M}{2 - \frac{I_{\mathrm{CP}}K_d}{C_f}}. \tag{47}$$

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] T.-H. Hsu, C.-C. Wang, and C.-Y. Lee, "Design and analysis of a portable high-speed clock generator," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 367–375, Apr. 2001.

[2] H. Tao, L. Tóth, and J. M. Khoury, "Analysis of timing jitter in bandpass sigma-delta modulators," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 991–1001, Aug. 1999.
[3] G. Chien and P. R. Gray, "A 900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1996–1999, Dec. 2000.
[4] 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 J. Solid-State Circuits*, vol. 36, pp. 417–423, Mar. 2001.
[5] C. Kim, I.-C. Hwang, and S.-M. Kang, "Low-power small-area ± 7.28 ps jitter 1 GHz DLL-based clock generator," in *Proc. SSCC Dig. Tech. Papers*, Feb. 2002, pp. 142–143.
[6] B. Kim, T. C. Weigandt, and P. R. Gray, "PLL/DLL system noise analysis for low-jitter clock synthesizer design," in *Proc. Int. Symp. Circuits and Systems*, London, U.K., June 1994.
[7] G. Chien, "Low-noise local oscillator design techniques using a dll-based frequency multiplier for wireless applications," Ph.D. dissertation, Univ. California, Berkeley, 2000.
[8] B. Nauta, "Analog CMOS low power design considerations," presented at the Low Power Workshop ESSCIRC Conf., Neuchâtel, Switzerland, Sept. 1996.
[9] F. Herzel and B. Razavi, "A study of oscillator jitter due to supply and substrate noise," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 56–62, Jan. 1999.
[10] J. G. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1723–1732, Nov. 1996.
[11] F. M. Gardner, "Charge-pump phase-locked-loops," *IEEE Trans. Commun.*, vol. COM-28, pp. 1849–1858, Nov. 1980.
[12] J. P. Hein and J. W. Scott, "z-domain model for discrete-time PLL's," *IEEE Trans. Circuits Syst. II*, vol. 35, pp. 1393–1400, Nov. 1988.
[13] C. S. Vaucher and D. Kasperkovitz, "A wide-band tuning system for fully integrated satellite receivers," *IEEE J. Solid-State Circuits*, vol. 33, pp. 987–997, July 1998.
[14] L. Lin, L. Tee, and P. R. Gray, "A 1.4 GHz differential low-noise CMOS frequency synthesizer using a wideband PLL architecture," in *Proc. ISSCC Dig. Tech. Papers*, Feb. 2000, pp. 204–205.
[15] T. C. Weigandt, B. Kim, and P. R. Gray, "Timing jitter analysis for high-frequency, low-power CMOS ring-oscillator design," presented at the Proc. Int. Symp. Circuits and Systems, London, U.K., June 1994.
[16] R. Farjad-rad *et al.*, "A 0.2–2 GHz 12 mW multiplying DLL for low-jitter clock synthesis in highly-integrated data communication chips," in *Proc. ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 76–77.
[17] S. Ye, L. Jansson, and I. Galton, "A multiple-crystal interface PLL with VCO realignment to reduce phase noise," in *Proc. ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 78–79.
[18] S. Kuboki, K. Kato, N. Miyakawa, and K. Matsubara, "Nonlinearity analysis of resistor string A/D converters," *IEEE Trans. Circuits Syst. II*, vol. CAS-29, pp. 383–390, June 1982.
[19] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1433–1440, Oct. 1989.
[20] A. Hastings, *The Art of Analog Layout*. Englewood Cliffs, NJ: Prentice-Hall, 2001.
[21] P. Larsson, "A simulator core for charge-pump PLLs," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 1323–1326, Sept. 1998.
[22] B. Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design*. Piscataway, NJ: IEEE Press, 1996.

Remco C. H. van de Beek (S'99) was born on September 3, 1974, in Wageningen, The Netherlands. He received the M.Sc. degree in electrical engineering from the University of Twente, Enschede, The Netherlands, in 1999. He then joined the Integrated Circuits Design Group at the same university, where he is currently working toward the Ph.D. degree in close cooperation with the Philips Research Laboratories, Eindhoven, the Netherlands.

His current research interests include gigahertz range low-jitter clock multiplication, PLL frequency synthesizers, and high-frequency logic circuits.

Eric A. M. Klumperink (M '98) was born on April 4, 1960, in Lichtenvoorde, The Netherlands. He received the B.Sc. degree from Hogere Technische School (HTS), Enschede, The Netherlands, in 1982 and the Ph.D. degree from the University of Twente, Enschede, The Netherlands, in 1997.

In 1984, he joined the Faculty of Electrical Engineering at the University of Twente, where he was mainly engaged in analog CMOS circuit design. Currently, he is an Assistant Professor and is involved in teaching and research at the IC-Design Laboratory, Faculty of Electrical Engineering, University of Twente, and the IC-Design Theme of the MESA+ Research Institute. His research interest is in CMOS circuits, especially for front-ends of integrated CMOS transceivers.

Cicero S. Vaucher (M'98–SM'02) was born in São Francisco de Assis, Brazil, in 1968. He received the electrical engineering degree from the Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, in 1989 and the Ph.D. degree in the same field from the University of Twente, Enschede, The Netherlands, in 2001.

Since 1990, he has been with Philips Research Laboratories, Eindhoven, The Netherlands, where he is currently a Senior Research Scientist in the Integrated Transceivers Department. He is the author of *Architectures for RF Frequency Synthesizers* (Norwell, MA: Kluwer, 2002) and is a coauthor of *Circuit Design for RF Transceivers* (Norwell, MA: Kluwer, 2001). He holds eight international patents on the subject of PLL and receiver design. His research activities have included implementations of low-power high-speed building blocks for PLL frequency synthesisers, synthesiser architectures for low-phase-noise and fast-settling-time applications, CAD modeling of PLL frequency synthesisers, and data/clock recovery and clock conversion circuits for optical transceivers. His current research involves analog IC design for microwave applications.

Bram Nauta (S'89–M'91) was born in Hengelo, The Netherlands, in 1964. He received the M.Sc. degree (*cum laude*) in electrical engineering and the Ph.D. degree in analog CMOS filters for very high frequencies from the University of Twente, Enschede, The Netherlands, in 1987 and 1991, respectively.

In 1991, he joined the Mixed-Signal Circuits and Systems Department of Philips Research, Eindhoven, The Netherlands, where he worked on high-speed A/D converters. From 1994 to 1998, he led a research group in the same department, working on "analog key modules." In 1998, he returned to the University of Twente, as a Full Professor heading the IC Design Group in the Department of Electrical Engineering, MESA+ Research Institute. He is also a part-time industry Consultant. His Ph.D. thesis was published as a book, *Analog CMOS Filters for Very High Frequencies* (Norwell, MA: Kluwer, 1993). He holds nine patents in circuit design. His current research interest is analog CMOS circuits for transceivers.

Dr. Nauta received the Shell Study Tour Award for his Ph.D. work in 1992. From 1997 to 1999, he served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–II: ANALOG AND DIGITAL SIGNAL PROCESSING and in 1998 he served as Guest Editor of IEEE JOURNAL OF SOLID-STATE CIRCUITS. In 2001, he became an Associate Editor of IEEE JOURNAL OF SOLID-STATE CIRCUITS. He is a Member of the Technical Program Committees of ESSCIRC and ISSCC.