Analytical model and behavioral simulation approach for a fractional-N synthesizer employing a sample-hold element

Cassia, Marco; Shah, Peter Jivan; Bruun, Erik

Published in:
IEEE transactions on circuits and systems - 2, Analog and digital signal processing

Link to article, DOI:
10.1109/TCSII.2003.819138

Publication date:
2003

Document Version
Publisher's PDF, also known as Version of record

Analytical Model and Behavioral Simulation Approach for a $\Sigma\Delta$ Fractional-N Synthesizer Employing a Sample-Hold Element

Marco Cassia, Peter Shah, *Member, IEEE*, and Erik Bruun, *Senior Member, IEEE*

***Abstract*—A previously unknown intrinsic nonlinearity of standard $\Sigma\Delta$ fractional-N synthesizers is identified. A general analytical model for $\Sigma\Delta$ fractional-N phased-locked loops (PLLs) that includes the effect of the nonlinearity is derived and an improvement to the synthesizer topology is discussed. Also, a new methodology for behavioral simulation is presented: the proposed methodology is based on an object-oriented event-driven approach and offers the possibility to perform very fast and accurate simulations, and the theoretical models developed validate the simulation results. We show a GSM example to demonstrate the applicability of the simulation methodology to real study cases.**

***Index Terms*—Linear systems, nonlinearities, phase-locked loops, phase noise, sigma–delta modulation, simulation.**

I. Introduction

THE DELTA–SIGMA modulation in fractional-N synthesizers is a technique that has been successfully demonstrated for high resolution and high-speed frequency synthesizers [1], [2]. These synthesizers use high-order multibit $\Sigma\Delta$ modulators [8] to dither the divider modulus, introducing the issue of high-frequency quantization noise down-folding. For this reason, the derivation of analytical models for noise analysis and the development of efficient techniques for fast and accurate simulations becomes very important.

Simulation of $\Sigma\Delta$ fractional-N synthesizers is difficult for many reasons [3]; simulation time tends to be long since a large number of samples is necessary in order to retrieve the statistical behavior of the system. The dithering applied on the divider modulus makes the behavior of the synthesizers nonperiodic in steady state; therefore, known methods for periodic steady-state simulations [6] cannot be applied to $\Sigma\Delta$ fractional-N synthesizers.

Traditional time sampling simulations based on fixed time-steps or adaptive time-steps quantize the location of the edges of the digital signals. This causes quantization noise and more severely, nonuniform sampling, which is a highly nonlinear phenomenon and leads to down-folding of high-frequency noise.

Different techniques to solve the quantization issue have been proposed in [3] and [5]. In [3], an area conservation principle approach allows to use uniform time-steps in the simulation. In [5], a simple event-driven approach is used in combination with iterative methods to calculate the loop filter response for integer-N phased-locked loops (PLLs). Event-driven simulators offer an alternative approach for simulating fractional-N synthesizers in a fast and accurate manner, and have so far been unexplored for this application area.

In this paper, we present and discuss a new simulation methodology based on an object-oriented event-driven approach [18]. This methodology, besides being accurate and highly efficient, prevents nonlinear time quantization from appearing in the simulation. In addition, it allows easy modification and augmentation of individual blocks separately without having to worry about interaction with other blocks.

Before discussing the simulation methodology, we identify a previously unknown intrinsic nonlinear phenomenon in the standard $\Sigma\Delta$ PLL topology [18], which causes down-folding of high-frequency quantization noise and hence increased close-in phase-noise. In Section II, we propose a simple enhancement to the synthesizer topology to eliminate the intrinsic nonlinearity; in Section III we derive a linear model, and in Section IV we extend the model to incorporate the nonlinear effect.

In Section V, we present and discuss the simulation methodology. Finally in Section VI, we compare results from simulations with the theory developed. Also we demonstrate the applicability of the simulation methodology to a direct GSM modulation synthesizer.

II. Sample-Hold Topology

Before deriving a linear model for $\Sigma\Delta$ fractional-N synthesizers, we address a nonlinear issue intrinsic to the standard $\Sigma\Delta$ synthesizer topology. The phase frequency detector samples the phase error in a nonuniform manner. The phase frequency detector produces UP and DOWN pulses of variable length occurring, respectively, after and before the sampling point. The sampling is thus spread out over time around the reference clock edge and that effectively constitutes nonuniform sampling. This is illustrated in Fig. 1.

Nonuniform sampling is a highly nonlinear phenomenon and causes the down-folding of high-frequency noise. The contribution of the down-folded noise to the overall output phase noise can be relevant, especially since high-frequency and high-power $\Sigma\Delta$ quantization noise is present.

Manuscript received May 1, 2003; revised July 2003. This work was supported by Qualcomm CDMA Technologies. This paper was recommended by Guest Editor M. Perrott

M. Cassia and E. Bruun are with the Ørsted–DTU Department, Technical University of Denmark, DK-2800 Lyngby, Denmark (e-mail: mca@oersted.dtu.dk).

P. Shah is with the RF Magic, San Diego, CA 92121 USA (e-mail: pshah@rf-magic.com).

Digital Object Identifier 10.1109/TCSII.2003.819138

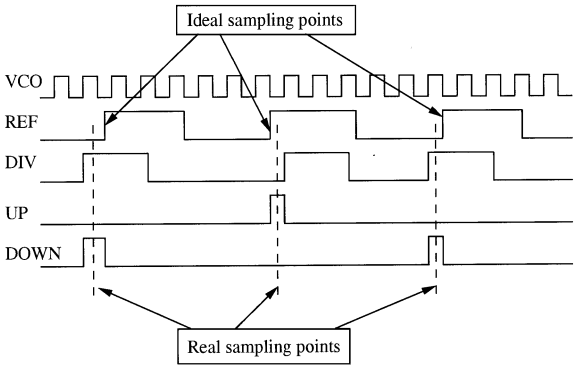


Fig. 1. Nonuniform sampling.

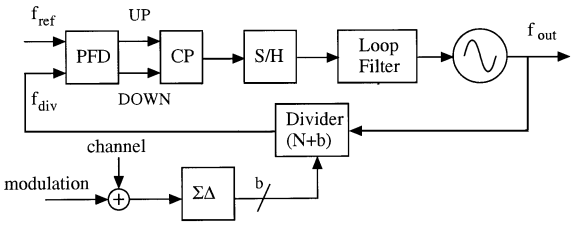


Fig. 2. S/H $\Sigma\Delta$ fractional-N synthesizer.

To solve the nonuniform sampling problem, we adopt the topology [18] shown in Fig. 2. The structure is similar to ordinary $\Sigma\Delta$ fractional-N synthesizers except for the presence of a sample-hold block between the charge-pump and the loop filter. By resampling the charge pump output at regular time intervals, the nonlinearity previously discussed is eliminated. The sample-hold has another beneficial effect: it prevents the modulation of the loop filter voltage by the reference clock, hence, ideally it eliminates reference spurs in the voltage-controlled oscillator (VCO) output. In reality, low-level spurs may appear at the output due to the charge feedthrough in the control switch.

The use of sample–hold detectors is known [12], [16] to give good spurious performance; sampled PLL circuits have been already used in clock and data-recovery circuits [13]. A sampled feed-forward network has been recently proposed in a clock generator PLL architecture [14]. However, to the knowledge of the authors, the sample-hold technique has not been used before in $\Sigma\Delta$ fractional-N synthesizers for the purpose of compensating the nonuniform sampling operation of the phase-frequency detector (PFD).

In Section III, we present a derivation of a linear model of the S/H $\Sigma\Delta$ fractional-N synthesizer. The resulting linear model is similar to [4], but the derivation is more straightforward and provides more intuitive insight.

III. Linear Model Derivation

The starting point is the sample-hold portion of the synthesizer. A possible implementation is shown in Fig. 3. This circuit uses a switched-capacitor integrator to carry out both the sample-hold function as well as the integrator function that is usually implemented by the loop filter. Note, that the sample-hold block is in series with the loop filter: both the integral and the proportional loop corrections are sampled and

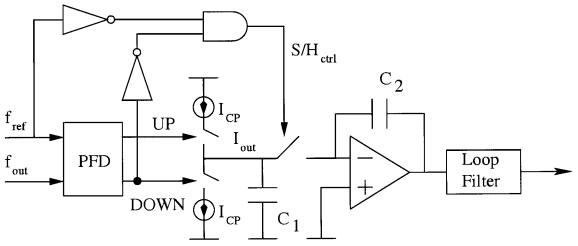


Fig. 3. Possible implementation of PFD and charge-pump with sample-hold.

held for each PFD sampling interval. To derive the transfer function we start by considering the charge deposited on the capacitance C_1

$$Q_{C_1} = \frac{\Delta\varphi(t)}{2\pi} T_{\mathrm{Ref}} \cdot I_{\mathrm{CP}} \tag{1}$$

where $\Delta\varphi(t)$ is the phase error waveform into the PFD. After a certain delay τ_{SH} the charge is transferred to C_2 and added to the charge previously stored

$$Q_{C_2}(t) = Q_{C_2}(t - T_{\mathrm{Ref}}) + Q_{C_1}(t - \tau_{\mathrm{SH}}) \tag{2}$$

In voltage terms and inserting the expression for Q_{C_1}

$$V_{C_2}(t) = V_{C_2}(t - T_{\mathrm{Ref}}) + \frac{I_{\mathrm{CP}}}{2\pi \cdot C_2} \cdot T_{\mathrm{Ref}} \cdot \Delta\varphi(t - \tau_{\mathrm{SH}}). \tag{3}$$

Taking the Laplace transform yields

$$\frac{V_{C_2}(s)}{\Delta\varphi(s)} = T_{\mathrm{Ref}} \cdot \frac{I_{\mathrm{CP}}}{2\pi \cdot C_2} \cdot \frac{e^{-s\tau_{\mathrm{SH}}}}{1 - e^{-sT_{\mathrm{Ref}}}}. \tag{4}$$

In (4), $V_{C_2}(s)$ is still modeled in the discrete-time domain, i.e. as a train of delta-functions. In reality, the output voltage is a staircase function. As a consequence, (4) is further modified by a zeroth-order hold network that converts the impulse-train into the staircase waveform. The transfer function of the zeroth-order hold network is given by

$$H_{\mathrm{ZOH}}(s) = \frac{1}{T_{\mathrm{Ref}}} \cdot \frac{1 - e^{-sT_{\mathrm{Ref}}}}{s}. \tag{5}$$

The actual transfer function from phase difference (PFD input) to integrator output is then given by

$$\frac{V_O(s)}{\Delta\varphi(s)} = H_{\mathrm{ZOH}}(s) \cdot \frac{V_{C_2}(s)}{\Delta\varphi(s)} = e^{-s\tau_{\mathrm{SH}}} \cdot \frac{I_{\mathrm{CP}}}{2\pi \cdot s \cdot C_2}. \tag{6}$$

Consequently, the circuit in Fig. 3 can be modeled as shown in Fig. 4. Note that in Fig. 4 the integration $1/sC_2$ has been absorbed in the loop filter transfer function $F(s)$. Thus, the only difference introduced in the linear model by the sample-hold is the delay τ_{SH}. Note that the sampling now always occurs at regular time intervals, namely at the negative edge of the reference clock.

In the setup shown in Fig. 3, the delay τ_{SH} is equal to half a reference period. The delay is necessary to allow the charge-pump current to be completely integrated before the sampling operation takes place. Note also that the sampling switch needs to be opened while the charge pump is active.

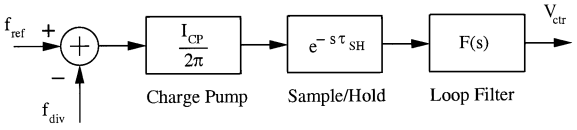


Fig. 4. Linear model of S/H portion.

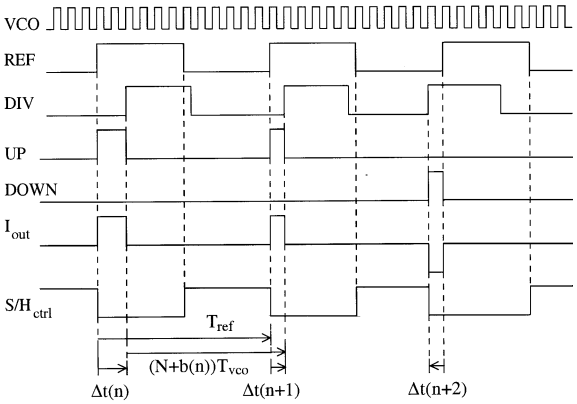


Fig. 5. PLL waveforms.

The control logic of Fig. 3 takes into account the fact that the rising edge of the DOWN pulse occurs before the rising edge of the reference clock.

If a trickle current is used in the charge-pump (e.g. only UP pulses are generated in the lock state) then it is sufficient to invert the reference clock signal to generate a proper $\mathrm{S/H_{ctrl}}$ signal.

A. Divider

We will now derive a simple linear model for the divider with dithering. The first step is to find the timing deviations with the aid of Fig. 5. *N* is the nominal divider modulus and $b(n)$ is the dithering value provided by the $\Sigma\Delta$ modulator. Note that the UP and DOWN pulses have variable length and occur, respectively, after and before the reference signal. As already stated, the sampling is spread out over time before and after the sampling point.

According to the timing diagram we can write

$$\Delta t(n+1) = \Delta t(n) + (N + b(n)) \cdot T_{\mathrm{VCO}} - T_{\mathrm{Ref}}. \tag{7}$$

Indicating with μ_b the average value of $b(n)$ (μ_b is the fractional divider value), the reference period T_{Ref} can be expressed as

$$T_{\mathrm{Ref}} = (N + \mu_b) T_{\mathrm{VCO}}. \tag{8}$$

In deriving (8), we are making the important approximation that T_{VCO} is constant. This assumption is reasonable for receive–transmit synthesizers with narrow modulation bandwidth. In these cases, the relative frequency variation of the VCO is small, which means that T_{VCO} is nearly constant.

Defining $b'(n) = b(n) - \mu_b$ and substituting T_{VCO} from (8) into (7) yields

$$\Delta t(n+1) = \Delta t(n) + \frac{T_{\mathrm{Ref}}}{N + \mu_b} b'(n). \tag{9}$$

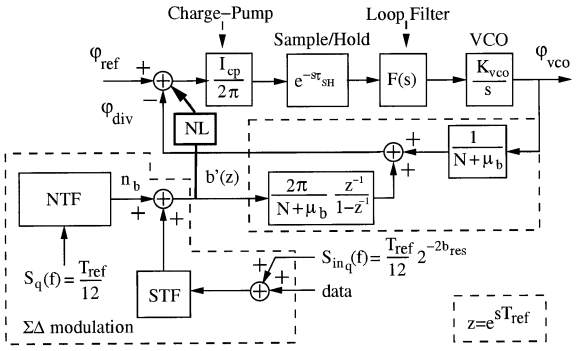


Fig. 6. Complete linearized sample–hold $\Sigma\Delta$ fractional-*N* PLL.

Converting to phase domain we have

$$\Delta\varphi = \frac{2\pi \Delta t}{T_{\mathrm{Ref}}}. \tag{10}$$

We can finally derive an expression for the additive noise caused by dithering the divider ratio

$$\Delta\varphi(n+1) = \Delta\varphi(n) + \frac{2\pi}{N + \mu_b} b'(n). \tag{11}$$

The Laplace transform yields

$$\Delta\varphi(s) = \frac{2\pi}{N + \mu_b} \cdot \frac{e^{-sT_{\mathrm{Ref}}}}{1 - e^{-sT_{\mathrm{Ref}}}} b'(z). \tag{12}$$

Setting $z = e^{sT_{\mathrm{Ref}}}$, (12) can be equivalently written in the digital domain (*Z*-transform)

$$\Delta\varphi(z) = \frac{2\pi}{N + \mu_b} \cdot \frac{z^{-1}}{1 - z^{-1}} b'(z). \tag{13}$$

The previous equation shows that the $\Sigma\Delta$ noise undergoes an integration but is otherwise shaped by the loop in exactly the same way as the reference clock phase noise.

The final linear model is shown in Fig. 6, where signal transfer function (STF) and the noise transfer function (NTF) are the $\Sigma\Delta$ modulator, respectively [8]. The NL block in the model indicates the nonlinear effect that occurs in the PLL if the sample-hold block is not used. An analytical derivation of such effect is presented in Section IV. The closed-loop transfer function $H_\theta(s)$ is given by (Fig. 6)

$$H_\theta(s) = \frac{\frac{I_{\mathrm{CP}}}{2\pi} e^{-s\tau_{\mathrm{SH}}} \cdot F(s) \frac{K_{\mathrm{vco}}}{s}}{1 + \frac{I_{\mathrm{CP}}}{2\pi} e^{-s\tau_{\mathrm{SH}}} \cdot F(s) \frac{K_{\mathrm{vco}}}{s} \frac{1}{N+\mu_b}}. \tag{14}$$

The phase noise properties can now be predicted from straightforward linear systems analysis [11]. Also, although Fig. 6 indicates $\Sigma\Delta$ modulation, the linear model has been derived with no assumption on the type of modulation used to dither the divider modulus (e.g., it is valid for any fractional-N topology [7]).

B. $\Sigma\Delta$ Modulation

The $\Sigma\Delta$ modulation can be modeled as additive phase contribution (also shown in Fig. 6). As an example, a $\Sigma\Delta$ MASH architecture [9] of order n is used in the analysis. The $\Sigma\Delta$ quantizer causes quantization noise n_b which is added to the output

word. Such noise is spread out over a bandwidth of $f_{\rm ref} = 1/T_{\rm ref}$ and is high-pass shaped by the $\Sigma\Delta$ modulator with a noise transfer function (NTF) given by

$$H_{\rm NTF}(z) = (1-z^{-1})^n\Big|_{z=e^{j2\pi fT_{\rm ref}}}. \tag{15}$$

The $\Sigma\Delta$ STF is given by

$$H_{\rm STF}(z) = (z^{-1})^n\Big|_{z=e^{j2\pi fT_{\rm ref}}}. \tag{16}$$

Assuming that the quantization noise is independent of the input signal, the power spectral density of the bit stream can be expressed as

$$S_{n_b}(f) = \frac{T_{\rm ref}}{12}\left|H_{\rm NTF}(f)\right|^2. \tag{17}$$

From the linear model of Fig. 6 we can find the transfer function from the output of the NTF to the output phase $\varphi_{\rm VCO}$

$$H_n(s) = \frac{2\pi}{N+\mu_b}\cdot\frac{e^{-sT_{ref}}}{1-e^{-sT_{ref}}}H_\theta(s) \tag{18}$$

Finally the output phase noise power spectral density (PSD) due to the $\Sigma\Delta$ quantization noise n_b is simply given by

$$S_{\varphi{\rm VCO}}(f) = \left|H_n(j2\pi fT)\right|^2 S_{nb}(f). \tag{19}$$

The effect of quantization at the $\Sigma\Delta$ input (i.e. due to finite input word length) can be evaluated in the same way. The PSD is given by

$$S_{\Sigma\Delta_{\rm in}}(f) = \frac{T_{\rm ref}}{12}\cdot 2^{-2b_{\rm res}}\cdot\left|H_{\rm STF}(f)\right|^2 \tag{20}$$

where $b_{\rm res}$ is the number of bits below the decimal point in the $\Sigma\Delta$ input. The calculation of the PSD of the PLL phase error due to the $\Sigma\Delta$ input quantization is then straightforward (Fig. 6)

$$S_{\varphi,\Sigma\Delta_{\rm in}}(f) = \left|H_n(j2\pi fT)\right|^2 S_{\Sigma\Delta_{\rm in}}(f). \tag{21}$$

The output phase noise due to other noise sources, such as charge-pump noise or VCO noise can be evaluated in a similar way.

IV. Analytical Evaluation of the Intrinsic Nonlinearity

As previously mentioned, in $\Sigma\Delta$ synthesizers an intrinsic non linearity affects the close-in phase noise. We will now show that in standard $\Sigma\Delta$ synthesizers, the charge-pump output $i_{\rm out}(t)$ contains an additional noise term, which is caused by the nonuniform pulse stretching shown in Fig. 5.

We begin by taking the Fourier transform of the charge-pump output

$$I_{\rm out}(f) = \int_{-\infty}^{\infty} i_{\rm out}(t)e^{-j2\pi ft}{\rm dt}. \tag{22}$$

With the aid of Fig. 5, the previous equation can be written as (23), shown at the bottom of page, which simplifies to

$$I_{\rm out}(f) = \sum_{n=-\infty}^{n=\infty}\int_{nT_{\rm REF}}^{nT_{\rm REF}+\Delta t(n)} I_{\rm CP}e^{-j2\pi ft}{\rm dt}. \tag{24}$$

By solving the integral, (24) becomes

$$I_{\rm out}(f) = I_{CP}\sum_{n=-\infty}^{n=\infty}\frac{-1}{j2\pi f}e^{-j2\pi fnT_{\rm ref}}\left(e^{-j2\pi f\Delta t(n)}-1\right). \tag{25}$$

We now perform a second-order Taylor series expansion of the $e^{-j2\pi f\Delta t(n)}$ term

$$\begin{aligned} I_{\rm out}(f) = {} & I_{CP}\sum_{n=-\infty}^{n=\infty}\frac{-1}{j2\pi f}e^{-j2\pi fnT_{\rm ref}} \\ & \cdot\Big[(1-j2\pi f\Delta t(n) \\ & -\frac{1}{2}\left(j2\pi f\Delta t(n)\right)^2\Big)-1\Big] \end{aligned} \tag{26}$$

$$\begin{aligned} I_{\rm out}(f) = {} & \frac{I_{CP}}{T_{\rm ref}}\left(T_{\rm ref}\sum_{n=-\infty}^{n=\infty}\Delta t(n)e^{-j2\pi fnT_{\rm ref}}\right) \\ & -j2\pi f\frac{I_{CP}}{T_{\rm ref}}\left(T_{ref}\sum_{n=-\infty}^{n=\infty}\frac{1}{2}\Delta t(n)^2\right. \\ & \left.\times e^{-j2\pi fnT_{\rm ref}}\right). \end{aligned} \tag{27}$$

Equation (27) contains two terms. The first one is simply a linearly filtered version of the quantization noise, as predicted by the linear model in the paper. The second term quantifies the undesired nonlinear effect caused by the nonuniform pulse stretching. As can be seen, it is essentially the Fourier transform of the filtered quantization noise squared, followed by a differentiation.

The NL block in Fig. 6 symbolizes the nonlinear effect and, according to the above analysis, it can be modeled as shown in Fig. 7.

Based on the previous analysis we can write an analytical expression for the PSD of the excess noise that occurs in standard $\Sigma\Delta$ PLL (i.e. without sample-hold)

$$\begin{aligned} S_{\theta{\rm out}_{\rm excess}}(f) = {} & 4\pi^2(2\pi f)^2\frac{1}{4} \\ & \times\left(S_{\Delta t}(f)\circledast S_{\Delta t}(f)\right)\left(\left|H_\theta(j2\pi fT)\right|^2\right) \end{aligned} \tag{28}$$

$$I_{\rm out}(f) = \begin{cases} \displaystyle\sum_{n=-\infty}^{n=\infty}\left(\int_{nT_{\rm REF}}^{nT_{\rm REF}+\Delta t(n)} I_{CP}e^{-j2\pi ft}dt\right), & \text{if } \Delta t(n) > 0 \\ \displaystyle\sum_{n=-\infty}^{n=\infty}\left(\int_{nT_{\rm REF}+\Delta t(n)}^{nT_{\rm REF}} -I_{\rm CP}e^{-j2\pi ft}dt\right), & \text{if } \Delta t(n) < 0 \end{cases} \tag{23}$$

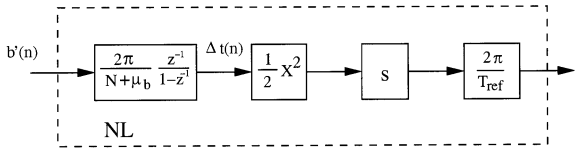


Fig. 7. Nonlinearity model.

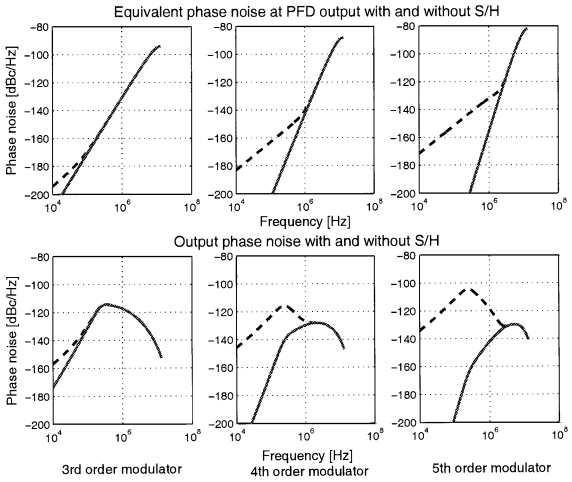


Fig. 8. Phase noise PSD for different $\Sigma\Delta$ modulator orders.

where "$\circledast$" denotes the convolution and $H_\theta(f)$ is given by (14) $S_{\Delta t}(f)$ is given by

$$S_{\Delta t}(f) = \left(\left| \frac{T_{\rm ref}}{N+\mu_b} \cdot \frac{1}{1-e^{-j2\pi f T_{\rm ref}}} \right| \right)^2 S_{n_b}(f) \qquad (29)$$

with $S_{n_b}(f)$ given by (17).

Fig. 8 shows the equivalent phase-noise at the phase-frequency detector input (top row) and at the PLL output (bottom row) for both sample-hold and nonsample-hold topology and for different $\Sigma\Delta$ modulator orders. The values of the parameters used in the graphics can be found in Section VI. If a nonsample-hold PLL is used then an excess noise appears and the total noise becomes as shown by the dashed curve. Of course, the regular $\Sigma\Delta$ quantization noise also gets worse with increasing frequency. So, at high-offset frequency the excess noise actually becomes insignificant in comparison with the $\Sigma\Delta$ noise. Notice also that the excess phase noise effect is more noticeable for high-order $\Sigma\Delta$ modulators. This is because the high-frequency quantization noise is stronger so that more noise is down-folded. On top of this, the low-frequency quantization noise is lower, which makes the excess noise more significant in comparison.

The contribution of the excess noise might not always be significant with respect to other PLL noise sources, such as the charge-pump noise, which usually dominates at low frequency. However it is still valuable to quantify and to model the effect of the nonlinearity in order to ensure correct performance of the PLL in all cases.

V. Event-Driven Object Oriented Methodology

As discussed in the introduction, the use of event-driven simulators is very attractive. Besides providing precise time-steps,

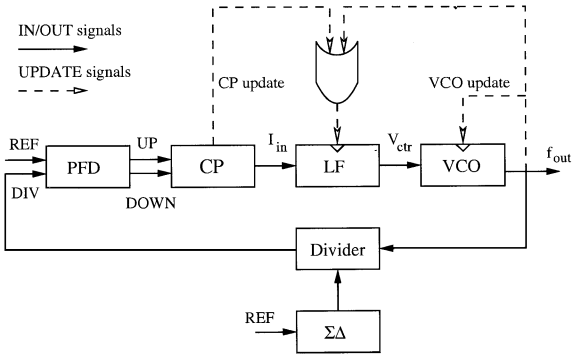


Fig. 9. Simulation model.

as explained later in the section, event-driven simulations are also very fast and highly efficient. In fact, the number of calculations is kept to a minimum because synthesizer signals and variables are calculated only when a transition occurs.

The simulation method proposed in [3] ensures extremely high computation speed because, instead of simulating the true time domain behavior, it effectively operates in a subsampled manner on the merged VCO-divider block. This idea makes the method in [3] very attractive too. However, this idea could equally well be used in the event-driven approach, speeding up the simulation tremendously. In this case, the VCO would sample the loop-filter once for every reference cycle. This subsampling operation implicitly relies on the assumption that the power level of the noise at high-frequency offset is not giving a significant contribution when aliased to low frequency. Thus, if the assumption holds, the event-driven approach would be equally as fast as the method in [3]. However, even without the VCO-divider merging approach, the event-driven method is already so fast that it is hardly worthwhile to use this merging technique.

A unique strength of the event-driven methodology we propose is that it is exact and does not require assumptions or approximations. The simulation setup is structured in an object-oriented way: PLL blocks are connected through signals that are responsible for timing and for data exchange, as shown in Fig. 9. Note that IN/OUT signals can operate also as implicit update signals (e.g. the UP/DOWN signals from the charge-pump). Whenever a block is called from the simulator, a specific operation is performed and an event may be posted. As shown in Fig. 10, the simulator inserts the event in the event queue in the proper time order and extracts from the queue the next event that needs to be executed, resulting in the update of the signals/variables of a block.

This means that each PLL block can be coded as an independent unit, without worrying about the interaction and the sequencing with the other blocks. The fact that each block is self-contained allows to change and refine the behavior of a single block without affecting the coding of the other PLL units. The simulator itself keeps track of the succession of the events with the event queue. A more detailed explanation of this concept can be found in [17].

The advantage of maintaining a simulation event-queue is that the simulation time points occur exactly at the moment of the execution of the event. Thus, the simulation time points are

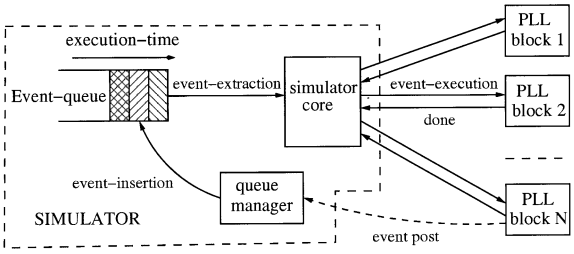


Fig. 10. Simulation structure.

always aligned with the edges of the signals, providing 100% accurate time-steps.

Coding the behavior of the synthesizer digital blocks is straightforward; the description of the loop filter and of the VCO requires particular attention, as discussed next. More details about the methodology implementation can be found in [17].

A. Loop Filter

We propose a simple method based on state–space equations description. The way the loop filter is modeled can be visualized with the help of Fig. 9. Every time the VCO and the charge-pump are executed, they post events requiring the update of the loop filter state. When these events are extracted from the event-queue to be executed, the simulator calls the loop filter to update its state and to calculate a new control voltage according to the actual input value. The event posted from the charge-pump indicates that a change has occurred at the loop filter input; the VCO event is posted to obtain the actual control voltage.

To describe the loop filter behavior in mathematical terms we start from its transfer function and we derive its State-Space Formulation. We assume the loop filter transfer function to be given by the following equation:

$$F(s) = \frac{1 + \frac{s}{z_0}}{sC \cdot \left(1 + \frac{s}{p_0}\right) \cdot \left(1 + \frac{s}{p_1}\right) \cdot \left(1 + \frac{s}{p_2}\right)}. \tag{30}$$

Note that (30) also includes the integrating capacitance. With a partial fraction expansion, (30) can be decomposed into four parallel blocks, namely an integrator and three first-order RC blocks. Noting that between the update times the input to the loop filter is constant (e.g. V_{in} is appearing as a staircase to the loop filter), the equation describing the behavior of each of the three RC blocks is given by (state equation solution)

$$V_x(t_1) = V_x(t_0) + (A_x V_{\text{in}}(t_0) - V_x(t_0)) \left(1 - e^{\frac{t_1 - t_0}{\tau_x}}\right). \tag{31}$$

The equation that describes the integrating block is given by

$$V_C(t_1) = V_C(t_0) + \frac{A_0 V_{\text{in}}(t_0)}{C}(t_1 - t_0). \tag{32}$$

The VCO control voltage is then given by

$$V_{\text{ctr}}(t) = V_1(t) + V_2(t) + V_3(t) + V_C(t). \tag{33}$$

The model for the loop filter is then simply given by a set of equations which describe exactly the behavior of the loop filter.

TABLE I
DESIGN PARAMETERS

f_{out}	N	μ_b	I_{CP}	K_{VCO}	τ_{SH}
$1907.75\ MHz$	73.375	0.375	$10\ \mu A$	$2\pi 100 MHz/V$	$0.5\ T_{REF}$

TABLE II
LOOP PARAMETERS

C_{CP}	z_1	ω_3	ω_4	ω_5
$34.522\ pF$	$2\pi \cdot 50kHz$	$2\pi \cdot 500KHz$	$2\pi \cdot 1MHz$	$2\pi \cdot 50MHz$

This representation of the loop filter can be directly converted into simulation code. It is important to underline that the filter behavior is modeled with no approximation. Also, the loop filter update takes place only when required by other blocks: the update time intervals are not uniform. This makes the simulation methodology very efficient, since the calculations occur only at the required time steps. Further implementation details can be found in [17].

B. VCO Model

The VCO is modeled as a self-updating block. Such operation can be visualized as shown in Fig. 9. The pseudocode describing the VCO behavior is presented in algorithm 1. The update takes place at discrete time instances, namely every half-VCO cycle. Every half-period the VCO receives the update VCO control voltage from the loop filter; on the basis of the received value, the new VCO period is calculated.

The VCO completes its execution by posting two events. The first event is the execution of the loop filter block at the next time point when the VCO update will take place. This ensures that the value used to calculate the semiperiod of the VCO is always updated. The second event is simply the scheduling of the next VCO block call.

Due to the finite number representation of the simulator, the effects of the number truncation represents a potential problem in the calculation of the VCO period. In order to avoid the accumulation of the truncation error, the calculation of the VCO semiperiod can be implemented as a first-order $\Sigma\Delta$ modulator. In this way, the accumulation error is always driven to zero on average.

Algorithm 1 VCO pseudocode

```
MODULE VCO
input control_voltage
output VCO_clk
  f_inst = f_FreeRUN + K_VCO · V_ctr // Update the in-
 stantaneous frequency
  VCO_semiperiod=0.5/f_inst // Calculate
 the new semiperiod
  VCO_clock = NOT (VCO_clock) // Update
 the VCO_clock signal
  POST_EVENT(update_loop @ cur-
 rent_sim_time + VCO_semiperiod)
  POST_EVENT(execute VCO @ cur-
 rent_sim_time + VCO_semiperiod)
END MODULE
```

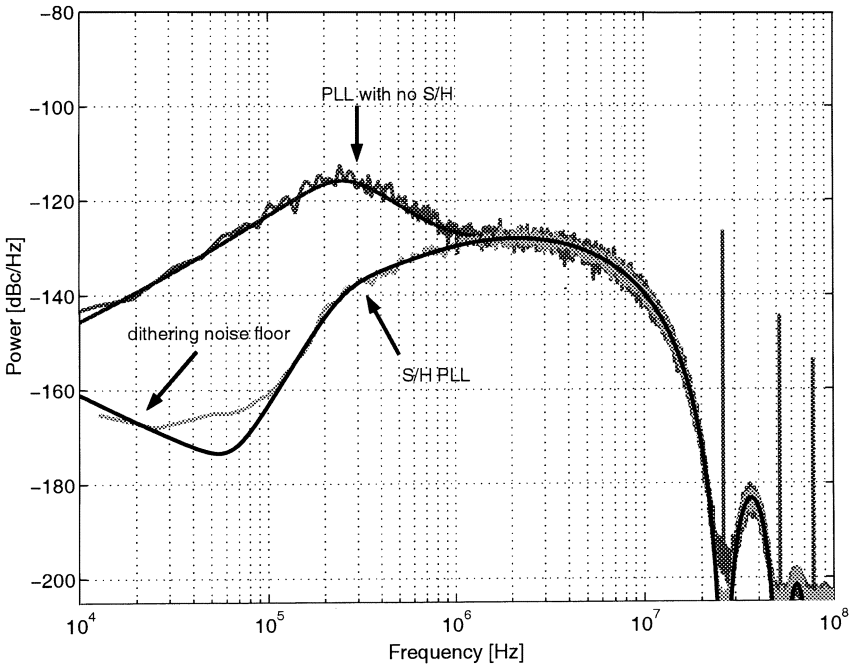


Fig. 11. Phase Noise PSD: sample-hold PLL versus regular PLL.

VI. Results

The PLL topology presented in Section II is simulated with Verilog XL, but the simulation methodology can be applied to any kind of event-driven simulators. For example, the simulation core can be easily implemented with a few lines of C code.

The choice of Verilog is a matter of convenience: its integration in the Cadence Environment allows easier debugging, schematic capture, and plotting capabilities. Moreover, the Cadence Environment offers the possibility to directly use the Verilog code together with Spice-like simulators to run mixed-mode simulation. However, simulations in a mixed-mode environment require long simulation time. As a comparison, to simulate in an event-driven simulation 2 million VCO cycles (equivalent to 1 ms) recording in a file 4 million data points, the time of execution is less than 15 min on a RISC 8500 processor (it reduces to only 5 min if the VCO simulation time points are not written to a file). The same simulation in a mixed-mode environment takes more than 20 h, without reaching the same accuracy. A fully analogue simulator such as SPICE would probably require a simulation time at least one order of magnitude longer.

The main parameters of the simulated PLL are resumed in Table I. The $\Sigma\Delta$ modulator is a MASH fourth order and the parameters of the loop filter are presented in Table II.

We now present several simulation results obtained by the event-driven methodology in order to

- validate the theory developed and evaluate the effect of the sample-hold block;
- evaluate the effects of nonidealities and nonuniform delay in the divider moduli;
- demonstrate the applicability of the simulation methodology to a real study case, namely direct GSM modulation.

We start by showing the effects of the nonuniform sampling at the PFD. The effect of other noise sources will be discussed later. Fig. 11 shows the PSD of the output phase noise φ_{VCO} due to the $\Sigma\Delta$ quantization for two different synthesizer topologies: the PSD of the sample-hold PLL is compared with the PSD of the standard PLL. The sample-hold PLL has a lower overall phase noise and does not present spurs. By contrast the standard PLL (i.e without sample-hold) has greatly increased close-in phase noise as well as reference spurs.

In the same figure, the PSD from simulations is compared with the predicted theoretical curves. Clearly, the curves obtained from the simulation match very well with the PSD described by (19) [for the sample-hold synthesizer topology] and (28) [for the standard synthesizer topology].

The low-frequency noise floor ("dithering noise floor" in Fig. 11) is due to a very small amount of dithering applied on the $\Sigma\Delta$ modulator input. In absence of modulated data, dithering is necessary to avoid the presence of fractional spurs.

In the previous figure, only the effect of the $\Sigma\Delta$ quantization noise on the output phase noise has been considered. The effects of other noise sources can be easily evaluated in the simulation, in a similar manner as described in [3]. Due to the object-oriented nature of the simulation it is easy to add new blocks that generate noise: the charge-pump white noise is obtained from a random number generator block and the VCO noise can be generated with another random number generator block followed by a filter block (coded in the same way as the loop filter). Another option is to read the noise data from a file; in this way it is possible to use data from other simulations or from real measurements. As an example, Fig. 12 shows the PSD of the output phase noise due to the contribution of $\Sigma\Delta$ quantization noise and VCO phase noise (in this example, the VCO phase noise is about -140 dBc/Hz @ 1-MHz offset). Together with the simulation result, Fig. 12 presents the predicted contribution of the single noise sources; the typical VCO phase noise (-20 dB/decade characteristic) determines an increased close-in phase noise.

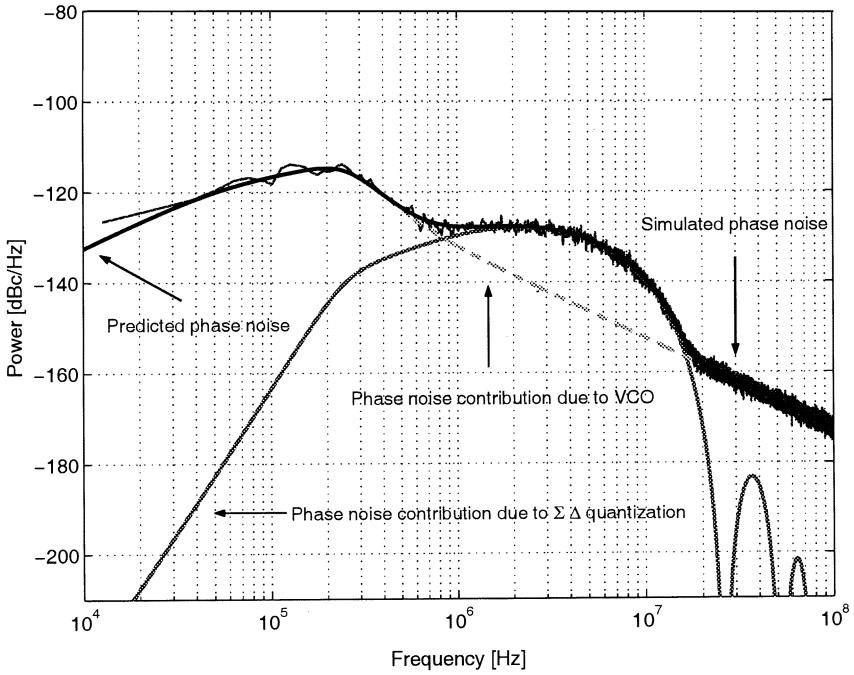


Fig. 12. Phase noise power spectral density with VCO noise added.

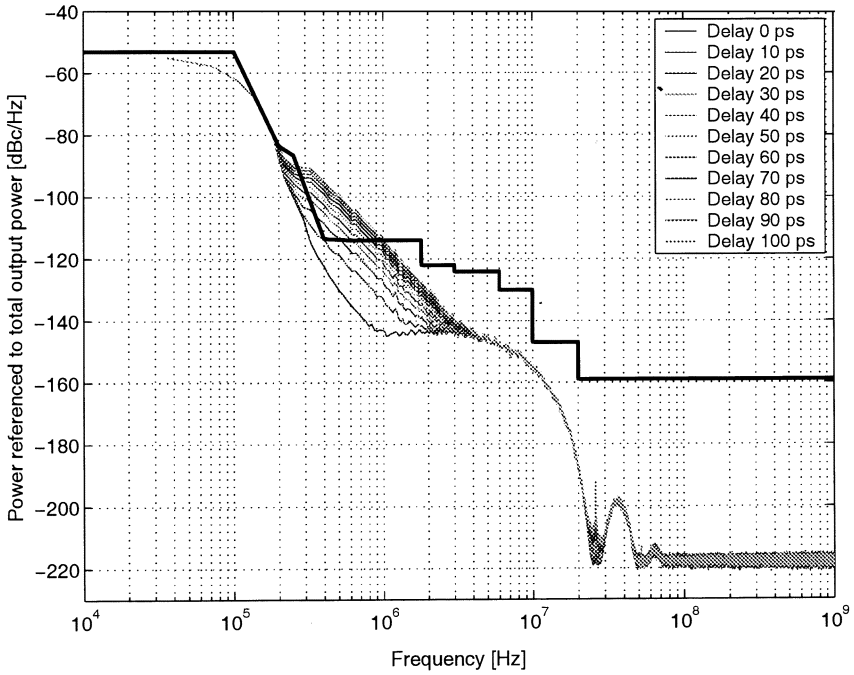


Fig. 13. Voltage PSD for different divider delays with GSM modulation.

A. Simulation Example: Direct GSM Modulation

The event-driven methodology and the linear model were applied to the study case of $\Sigma\Delta$ synthesizers for direct GSM modulation. The effects of nonidealities such as charge-pump mismatches, variation in the VCO gain, and variable delay in the divider modulus can be easily evaluated with the aid of the simulations. A brief account of the results will be given here; more results can be find in [17].

To evaluate the dynamic behavior of the simulator real GSM data was fed into the $\Sigma\Delta$ modulator through a digital prewarp filter [15], which compensates for the PLL transfer function. The transmitted output spectrum lies within the mask specified by the GSM standard and the rms phase error is smaller than 0.5° rms in the ideal condition.

As an example, the effects of a variable delay on a single divider modulus can be seen in Fig. 13. When the delay increases, the transmit power spectrum lies outside the mask specification. In fact, nonuniform propagation delay for the divider moduli is equivalent to nonuniform quantization in multibit $\Sigma\Delta$ DACs and causes down-folding of high-frequency noise.

The small reference spur in Fig. 13 is caused by a small dc content in the input data. In fact, the input data of the $\Sigma\Delta$ modulator is not ideal, but it is taken from a real implementation (e.g., the length of the Gaussian filter is finite).

The conclusions from the simulation on the study case can be summarized as follows.

- Identical results are achieved with a $\Sigma\Delta$ modulator based on a MASH or on a Candy architecture [10].
- It is important to ensure equal propagation delay for all divider moduli, otherwise the transmit power will exceed the GSM mask specification
- Even a small mismatch in the charge-pump currents results in a large close-in phase noise increase. To compensate the charge-pump current mismatches a fixed trickle current source can be used, in order to have pulses in only one direction (e.g. only UP pulses) under lock condition. The penalty of this choice is an increased spur level in the output spectrum for the standard PLL, but not for the sample-hold PLL.
- For receive synthesizers, the sample-hold topology greatly reduces close-in phase noise. In transmit mode, the increased close-in phase-noise integrates up to a relatively small rms phase error; consequently, it is acceptable to use the standard topology. However, the sample-hold eliminates the spur problems; this means that a trickle current can be used in the charge-pump to compensate for current mismatches.

The same conclusions are obtained in the study of a $\Sigma\Delta$ synthesizer whose target is the DCS specification. This indicates that the sample-hold PLL is suitable for both direct GSM/DCS modulation.

VII. Conclusion

This work identified an intrinsic nonlinearity of standard $\Sigma\Delta$ synthesizers and presented a sample-hold topology to solve this issue. The sample-hold also eliminates the problem of reference spurs in the output spectrum. A general analytical model was derived for the $\Sigma\Delta$ fractional-N synthesizer and was augmented to include the effects of the discussed nonlinearity. Moreover the model is valid for any kind of divider dithering, not just $\Sigma\Delta$ modulation; thus, regular fractional-N PLLs can also be analyzed using this model.

We also proposed a new simulation approach based on a object-oriented event-driven methodology. The simulation methodology is very accurate because it does not require approximations and undesirable time quantization phenomena are avoided, the only limit being the numerical accuracy of the event-driven simulator. One of the advantages of this approach it is its capability to naturally predict nonobvious phenomena such as noise down-folding, without having to resort to any special measures.

The comparisons presented in Section VI demonstrate a very good match between the theoretical model and the simulations. The examples provided show that the simulation methodology can be applied to the study of the effects of multiple nonidealities. As an example, a study case for direct GSM/DCS modulation was briefly presented and a summary of the results was shown, which indicate that the sample–hold $\Sigma\Delta$ fractional-N synthesizer is suitable for fulfilling the GSM/DCS standard.

Acknowledgment

The authors thank the QCT Department of Qualcomm CDMA Technologies for the valuable help and support in this work, and also thank P. Andreani for insightful discussions and the reviewers for their useful comments.

References

[1] T. A. Riley, M. A. Copeland, and T. A. Kwasniewski, "Delta–sigma modulation in fractional-*N* frequency synthesis," *J. Solid-State Circuits*, vol. 28, pp. 553–559, May 1993.

[2] T. Kenny, T. Riley, N. Filiol, and M. Copeland, "Design and realization of a digital delta–sigma modulator for fractional-N frequency synthesis," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 510–521, Mar. 1999.

[3] M. H. Perrott, "Fast and accurate behavioral simulation of fractional-N frequency synthesizers and other PLL/DLL circuits," in *Proc. Design Automation Conf. (DAC)*, June 2002, pp. 498–503.

[4] M. H. Perrott, M. D. Trott, and C. G. Sodini, "A modeling approach for $\Sigma-\Delta$ fractional-N frequency synthesizers allowing straightforward noise analysis," *J. Solid-State Circuits*, vol. 37, pp. 1028–1038, Aug. 2002.

[5] A. Demir, E. Liu, A. L. Sangiovanni-Vincentelli, and I. Vassiliou, "Behavioral simulation techniques for phase/delay-locked systems," in *Proc. Custom Integrated Circuits Conf. (CICC)*, 1994, pp. 453–456.

[6] K. Kundert, J. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*. Norwell, MA: Kluwer, 1990.

[7] B. Razavi, *RF Microelectronics*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[8] D. Johns and K. Martin, *Analog Integrated Circuit Design*, New York: Wiley, 1997.

[9] J. Candy and G. Temes, *Oversampling Delta–Sigma Data Converters*. Piscataway, NJ: IEEE Press, 1992.

[10] J. C. Candy, "A use of double integration in sigma delta modulation," *IEEE Trans. Commun.*, vol. COM-33, pp. 254–258, Mar. 1985.

[11] V. F. Kroupa and L. Sojdr, "Phase-lock loops of higher orders," in *Proc. 2nd Int. Conf. Frequency Control and Synthesis*, 1989, pp. 65–68.

[12] U. L. Rohde, *Digital PLL Frequency Synthesizers*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[13] N. Ishihara and Y. Akazawa, "A monolithic 156 Mb/s clock and data recovery PLL circuit using the sample-and-hold technique," *J. Solid-State Circuits*, vol. 32, pp. 1566–1571, Dec 1997.

[14] J. G. Maneatis, J. Kim, I. McClatchie, and J. Maxey, "Self-biased high-bandwidth low-jitter 1-to-4096 multiplier clock generator PLL," in *Proc. DAC*, 2003, pp. 688–690.

[15] M. H. Perrott, T. L. Tewksbury, and C. G. Sodini, "A 27 mW CMOS fractional-*N* synthesizer using digital compensation for 2.5 Mbit/s GFSK modulation," *J. Solid-State Circuits*, vol. 32, pp. 2048–2060, Dec. 1997.

[16] J. A. Crawford, *Frequency Synthesizer Design Handbook*. Norwood, MA: Artech House, 1994.

[17] *http://www.oersted.dtu.dk/personal/mca/oos.html* [Online]

[18] M. Cassia, P. Shah, and E. Bruun, "A spur-free fractional-N $\Sigma\Delta$ PLL for GSM applications: linear model and simulations," in *Proc. ISCAS*, 2003, pp. 1065–1068.

Acknowledgment

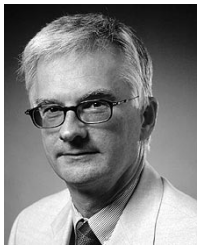
The authors thank the QCT Department of Qualcomm CDMA Technologies for the valuable help and support in this work, and also thank P. Andreani for insightful discussions and the reviewers for their useful comments.

Marco Cassia was born in Bergamo, Italy, in 1974. He received the M.Sc. degree in engineering from the Technical University of Denmark, Lyngby, in 2000, and the M.Sc. degree in electrical engineering from Politecnico di Milano, Italy, in July 2000. He is currently working toward the Ph.D. degree at the Technical University of Denmark, Lyngby.

From July 2001 to July 2002, he was with the QCT Department of Qualcomm CDMA Technologies, San Diego, CA, working with direct modulation synthesizers. His main research interests include low-power low-voltage RF systems.

Peter Shah (M'89) was born in Copenhagen Denmark, in 1966. He received the M.Sc.E.E. and Ph.D degrees from The Technical University of Denmark, Lyngby, in 1990 and 1993, respectively.

From 1993 to 1995, he was a Post-Doctoral Research Assistant with the Imperial College in London, England, where he worked on switched-current circuits. In 1996, he joined PCSI, San Diego, CA, (which was subsequently acquired by Conexant Systems, San Diego, CA) as an RFIC Design Engineer, working on transceiver chips for the PHS cellular phone system. In 1998, he joined Qualcomm, San Diego, CA, where he worked on RFICs for CDMA mobile phones and for GPS. In December 2002, he joined RFMagic, San Diego, CA, where he is currently working on RFICs for consumer electronics. His research interests include RFIC architecture and design, including sigma-delta PLLs, A/D, D/A converters, LNAs, mixers, and continuous-time filters.



Erik Bruun (M'72–SM'02) received the M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Denmark, Lyngby, in 1974 and 1980, respectively, the B.Com. degree from the Copenhagen Business School, Denmark, in 1980, and the Dr.Techn. degree from the Technical University of Denmark in 2000,

In 1974, and again, from 1980 to 1984, he was with Christian Rovsing A/S, Denmark, working on the development of space electronics and test equipment for space electronics. From 1974 to 1980, he was with the Laboratory for Semiconductor Technology, Technical University of Denmark, working in the fields of nMOS memory devices, I^2L devices, bipolar analog circuits, and custom integrated circuits. From 1984 to 1989, he was Managing Director of Danmos Microsystems ApS, Denmark. Since 1989, he has been a Professor of analog electronics with the Technical University of Denmark, where from 1995 to 2001, he served as Head of the Sector of Information Technology, Electronics, and Mathematics. Since 2001, he has been Head of Ørsted • DTU. His current research interests include RF integrated circuit design and integrated circuits for mobile phones.