An HDL Model of a Digitally Controlled Oscillator for Rapid Digital PLL Prototyping

V. Milovanović and B. Nikolić

Abstract— **Simulating digitally-controlled oscillators (DCOs) presents a challenging trade-off between the accuracy and the speed. Simplistic DCO simulations have been pursued, but had limitations with nonideal effects. Here, an event-driven behavioral model of a DCO which is able to capture arbitrary oscillating frequency tuning curves and to support multiple codeword inputs with varying granularity is presented. The DCO description provides supply voltage and ambient temperature dependencies, as well as the corner tests and random fluctuations for full PVT variability analysis. It is further equipped with thermal and flicker noise models. This fully-featured HDL implementation of a DCO is intended for use in a discrete-event simulation environment along with the synthesizable logic for rapid digital PLL analysis. A usual design case of a clock generator demonstrates the possibility of subtle random noise versus limit cycle optimization in a quick and convenient manner.**

I. Introduction

Deep nanometer CMOS processes favor digitally intensive realizations of traditional analog circuitry. As CMOS technology advanced both in downscaling and complexity, the burden of performance maintenance and adoption of necessary analog components into newer process nodes quickly overweighted their advantages. Typical example are phased-lock loop architectures which in the past decade exhibited an abrupt shift towards fully digital implementations [1–4]. Nowadays, digital PLLs are dominant not only in clock generation but also in the domain of RF frequency synthesis [1].

More recently, partially [2] and even completely [3] synthesizable DPLLs emerged as an alternative for applications where strict loop bandwidth is not required. These somewhat looser scenarios in which jitter and not precise phase noise level is of paramount importance, allowed replacement of power-hungry time-to-digital converters (TDCs) with rather simple bang-bang (1-bit) binary phase detectors [4] and/or pulse injectors [2, 3].

Area efficiency of digital loop filters is primarily attributed to elimination of bulky analog filter capacitors. Furthermore, they are easily tailored to different needs by soft programming. Finally, the synthesizable PLL topologies rely on standard-cell libraries and commercial EDA tools, thereby considerably shortening design cycle and cost. All these facts promote digital PLLs to design of choice in modern deeply-scaled technologies.

V. Milovanović is with the Faculty of Engineering, University of Kragujevac, Sestre Janjić 6, Kragujevac, (e-mail: vlada@kg.ac.rs)

B. Nikolić is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2108 Allston Way, Berkeley, CA 94704, (e-mail: bora@eecs.berkeley.edu).

Notwithstanding, the endeavor towards digital PLLs requested from an oscillator to be a digitally-controlled device, instead. A couple of distinct design patterns in contemporary digitally-controlled oscillators (DCOs) can be observed. Firstly, a hybrid approach, in which a conventional analog voltage-controlled oscillator is preceded by a digital-to-analog converter (DAC) [2], and secondly, direct digital-to-frequency conversion [5]. In either case, the oscillator still suffers from random noise variations which define its open-loop characteristics.

Noise sources in every DCO may be divided into truly random, such as thermal and flicker noise, and deterministic, coming from finite quantization error. And although the latter can be well-handled by discrete-event simulators, continuous-time SPICE-like circuit models are mandatory in analyzing the contribution of the former ones. It greatly limits digital PLL design efficiency which is anyway considered cumbersome and slow.

The oversimplified DCO models, that do not include colored noise sources, simply prove to be inadequate [4] for the task of an overall digital PLL design. Depending on the time constants that differ by several orders of magnitude, transient simulations of the complete PLL, which contains imported digital netlists in an analog environment, might take weeks to verify locking process.

Fast bit-accurate analog and digital co-simulations [6] sound appealing especially for system-level design and optimization, but involve yet another tool, further complicating already quite complex digital PLL design flow.

This paper presents a hardware-description language (or HDL-compatible) model of a DCO that may be implemented either in Verilog or any high-level synthesis language. Besides the basic frequency tuning curve it also features extensions for PVT corner/variability and noise simulations thus enabling a quick system-level optimization and possibility of rapid DPLL prototyping.

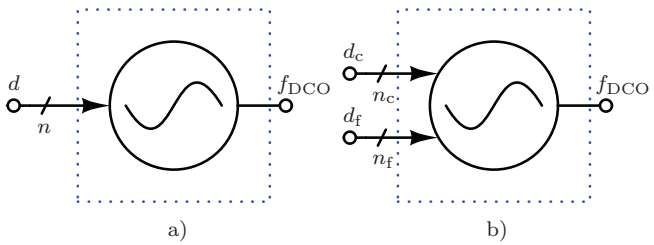

Fig. 1. DCO in which the oscillating frequency f_{DCO} depends on a) single, and b) coarse and fine, multibit control codes d.

978-1-5386-2563-7/17/$31.00 ©2017 IEEE

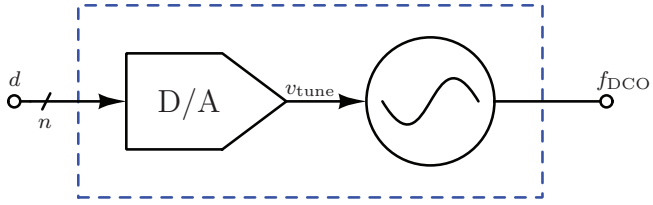

Fig. 2. DCO simply formed by cascading a DAC and a VCO.

II. DCO Frequency vs Code Tuning Curves

Undoubtedly, the most critical component of any digital PLL system is a digitally-controlled oscillator. DCO's oscillating frequency/period at its output is proportional to the digital code word on its input(s). This frequency/period dependence on the input data does not have to be linear. A single and more multibit inputs are equally common, as illustrated by Fig. 1 symbols. Practical DCOs seldom have linear dependence and use only one n-bit wide input bus. Regardless, linear oscillating frequency or period dependence as expressed by:

$$f_{\text{DCO}} = f_0 + K_{\text{F}} \cdot d \quad \Longleftrightarrow \quad T_{\text{DCO}} = T_0 + K_{\text{T}} \cdot d$$

in which f_0/T_0 and $K_{\text{F}}/K_{\text{T}}$ represent the free-running frequency/period of the DCO and its frequency/period linear sensitivity on the input data d are very useful.

It should be noted that the equation for the oscillating period directly translates to HDL's parametric delay control (e.g., `#<time> <statement>;` in Verilog). Nonetheless, simple relationship between the oscillating frequency and period $T_{\text{DCO}} = 1/f_{\text{DCO}}$, enables convenient use of either DCO quantity for its description.

Linear frequency and period dependencies give an important insight into the ratio between the sensitivities at some particular oscillating point. It can be written:

$$K_{\text{F}}/K_{\text{T}} = f_{\text{DCO}}/T_{\text{DCO}} = f_{\text{DCO}}^2 = 1/T_{\text{DCO}}^2 \quad .$$

The previous expression provides means to switch between instantaneous oscillating frequency and period of oscillation even for nonlinear oscillator tuning curves.

In cases where independent coarse/fine digital tuning inputs are available, the oscillating frequency or period may be coded using additional sensitivity parameters:

$$f_{\text{DCO}} = f_0 + K_{\text{Fcoarse}} \cdot d_{\text{coarse}} + K_{\text{Ffine}} \cdot d_{\text{fine}} \quad .$$

It usually makes sense when physically independent DCO tuning mechanisms are used, like for example, regulating ring oscillator's supply voltage rail for coarse and capacitive tap load or current through starved inverters for fine grain frequency or period modulation.

To account for potential nonlinearities in, let's say, hybrid DCOs of Fig. 2, the introduction of quadratic, cubic and even higher order terms in general is just as easily accomplished to fit any tuning curve. Exploiting look-up table as the universal modeling instrument can also be utilized for rather complex dependency curves.

III. DCO Corner Cases and Variability Model

The previously outlined digitally-controlled oscillator model did not account for different process (P) corners, supply voltage (V) nor temperature (T) dependence. Model extensions that involve PVT variability are critical in functional verification as they can increase yield.

A. DCO Frequency as a Function of Process Corners

Presumably the easiest way to include various process corners into the proposed HDL oscillator model is through compiler directives and parametrized modules.

In this manner, for example, five front end of line (FEOL) process corners, in particular typical-typical (TT), fast-fast (FF), slow-slow (SS), fast-slow (FS), and slow-fast (SF) would merely correspond to five sets of DCO parameters and several preprocessor directives.

Similarly, adding the back end of line (BEOL) technology corners, or any other kind of corners, indeed reduces down to analogous SPICE simulation followed by the extraction and inclusion of the free-running oscillating frequency/period and its input code sensitivities.

B. Supply Voltage Dependence of the DCO Frequency

Even though digitally- and voltage-controlled oscillators are almost exclusively supplied through a linear regulator to prevent power supply noise coupling, it is occasionally beneficial to include this indicator, too.

As a matter of fact, equipping the provided model with the dependence on supply voltage V_{DD} does not differ from what has already been done with input d, apart from supply pin's analog nature. Even if creating a floating-point port in an HDL of choice (e.g., plain Verilog) is not supported, it is trivial to circumvent.

It should be stressed here that the very same methodology can be applied to build an HDL model of a VCO.

C. Temperature Dependence of the DCO Frequency

In practice, output frequency of every oscillator operating in an open loop will display certain temperature dependence. This temperature instability is eventually compensated in the negative feedback loop. In order to model temperature-related frequency drift, another real DCO module input port is allocated and the actual ambient temperature is administered from the testbench.

Internally, the free-running and sensitivity DCO parameters would not stay constant, rather they will be a function of this floating-point input. There are no limitations on the temperature dependency function, but straightforward polynomial fit with the coefficients extracted from adequate SPICE simulations often suffice.

For linear DCO curve, this is analytically expressed:

$$f_{\text{DCO}}(d, T) = f_0(T) + K_{\text{F}}(T) \cdot d \quad ,$$

where T is the ambient temperature. The given model enables realistic prediction of frequency vs temperature.

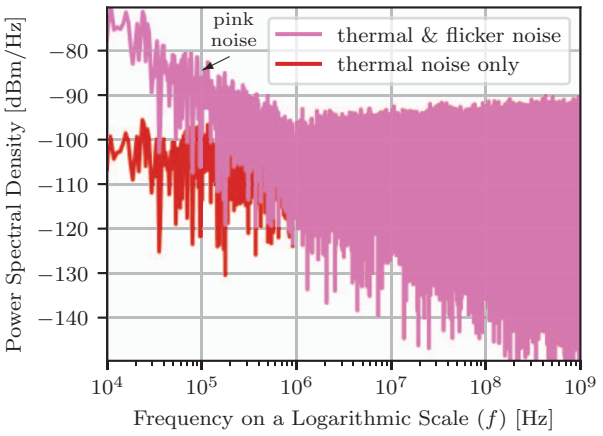

Fig. 3. Generated noise power spectra for the oscillator model.

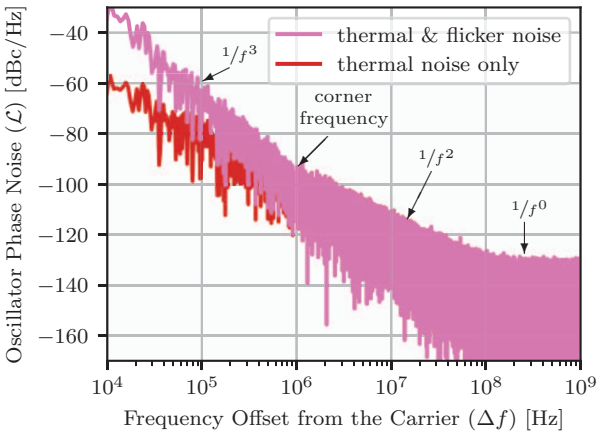

Fig. 4. Simulated phase-noise spectra with the oscillator model.

D. Random Variability of the DCO Tuning Curve

Yet another crucial component for predicting and improving manufacturing yields in integrated circuit (IC) design are statistical simulations. Perhaps the simplest form of randomness that could be brought into a model is by picking its parameters from a predefined probability distribution. That is to say, every parameter of the presented oscillator model is sampled from the normal distribution (incorporated in Verilog HDL through `$dist_normal()` system call/function). The expected parameter value (mean) coincides with the nominal corner while its standard deviation is drawn from results of the corresponding SPICE-level Monte-Carlo simulations. Ultimately, the DCO model parameters are taken from these distributions before digital simulation is ran.

IV. Oscillator Phase Noise Modeling

An ideal oscillator concentrates all of its power in a single frequency. Real oscillators, however, spread their power also into the nearby frequencies. This spectral frequency spreading is often referred to as phase noise.

Power spectral density of computer-generated pseudorandom noise used in the proposed HDL model and the simulated phase-noise spectrum of a typical oscillator are respectively given in Fig. 3 and Fig. 4 as log-log graphs. The latter one is ordinarily normalized to dBc/Hz and plotted against the offset frequency Δf from the carrier. The phase noise frequency profile commonly traverses from the $1/f^3$ over $1/f^2$ down to $1/f^0$ slope regions. The part in the middle is the so-called thermal noise region because it is induced by white or uncorrelated timing variations in the period of oscillation. It is actually the up-converted Gaussian noise and generally represents the dominant noise mechanism in oscillators [1]. The $1/f$ flicker noise of electronic devices is also substantial particularly for lower offset frequencies where after upconversion it appears as $1/f^3$ region. Finally, the flat-band region is the thermal electronic noise added outside the oscillator, such as in an output buffer, and does not affect the oscillator's time base.

Oscillator perturbations observable in the frequency domain have the underlying cause in the time domain, where the exact time of one oscillating period differs from another. This timing error variance is called jitter.

A normally distributed random variable with zero mean added to the oscillating period value would act as period jitter. In the phase noise power spectral density (PSD) plot it will mimic the thermal noise region featuring the $1/f^2$ slope. To avoid potential spurious tones, a high quality pseudorandom number generator (PRNG) with long repetition periods, such as the widely available standard Mersenne Twister, should be employed.

Contrary to the white one, the colored noise is more challenging to synthesize. Even though approximative time-domain methods for generating pink ($1/f^\alpha$) noise do exist [7], the artificially created noise with arbitrary frequency shape and accuracy can only be retrieved by specifying its spectrum. More specifically, in this implementation, the noise sequence values are a product of an inverse (Fast) Fourier Transform (FFT) of spectral coefficients properly scaled within the frequency domain.

Namely, a temporal noise sequence featuring whatever noise spectrum profile (including the flicker's $1/f$) may be obtained by generating the appropriate noise coefficients in spectral space. The magnitudes of the coefficients should be chosen to produce the desired spectrum shape while phases should be taken randomly.

Following the inverse FFT with the upper coefficients as argument yields the wanted noise sequence values. A slight caveat to this method is that precise specification of the coefficient magnitudes would not be strictly correct (as they should also be random due to the stochastic nature of the process). To fabricate a faithful pink noise time series, firstly the discrete Fourier transform (DFT) of time domain white noise is computed to produce its spectrum. This spectrum is then shaped according to the noise color flavor ($1/f^\alpha$) and its parameters (α and the corner frequency f_c). Lastly, the IFFT gives desired sequence with the disadvantage that maximum simulation time needs to be known beforehand.

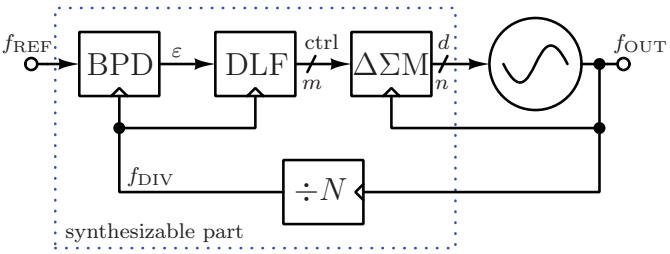

Fig. 5. Simplified block diagram of a bang-bang digital PLL.

V. HDL-Based Digital PLL Design Example

Although noiseless oscillator models can be of use in analyzing cases when period quantization errors introduced by the concurrent presence of phase detector and finite resolution DCOs are much larger than random-noise fluctuations, the real difference is made by fully featured oscillator noise model. Now not only the spurious tone locations determined by the limit cycle frequencies and caused by quasi-periodic orbits in the state space can be obtained, but also their magnitudes.

For the sake of demonstrating the full capability of the presented DCO model a conventional design procedure [4] was carried out for a study case of a regular digital PLL clock generator for system on a chip which contains large synchronous digital fabric accompanied with mixed-signal circuitry like data converters. On the account of a fairly relaxed application in which the absolute jitter is the key metric, a bang-bang type-II second-order integer-N digital PLL of Fig. 5 was selected. To further alleviate demands on the DCO's frequency resolution a MASH 1-1 digital $\Delta\Sigma$ modulator was used as fractional three-bit dithering signal.

System-level digital PLL specifications also included:

- $f_{REF} = 125\,\text{MHz}$ – input reference clock frequency;
- $f_{OUT} = 2\,\text{GHz}$ – synthesized output clock frequency;

with RMS absolute and period jitter values below 1 ps. The above clock synthesizer requirements can be satisfied with DCO oscillating frequency gain of approximately $K_F = 3.14\,\text{MHz/bit}$ which @2 GHz translates to period sensitivity of $K_T = 0.785\,\text{ps/bit}$. These numbers can be even achieved [5] with a ring-based oscillator.

The rest of the sub-blocks in this DPLL are described on a behavior RTL, synthesized from it and automatically placed & routed in a well-established digital flow.

With the free running DCO period jitter of $\approx 0.12\,\text{ps}$ and the corner knee frequency $f_c = 1\,\text{MHz}$, which are obtained from SPICE simulations, the fully equipped DCO model is built. This model is used along-side HDL description of the remaining DPLL modules in an event-driven simulator for system optimization & verification.

The ability of achieving the frequency and phase lock under all conditions is proved. DPLL's phase noise in the sweet spot of the trade-off between random and quantization noise is shown in Fig. 6. Integrated PSD of the PLL yields jitter values that meet the specifications.

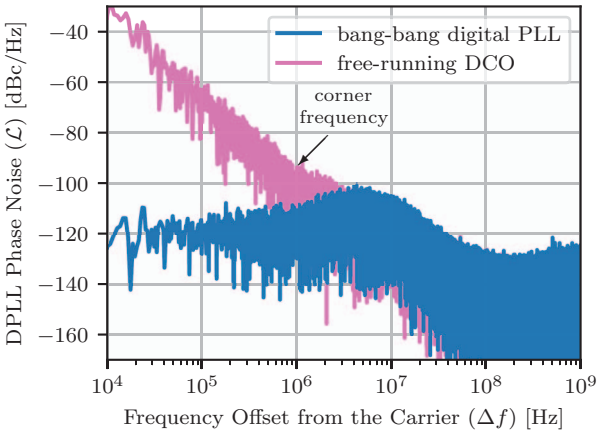

Fig. 6. Phase-noise of the optimized digital PLL clock generator.

VI. Conclusions

A complete DCO model intended for digital PLL design and ready to use inside discrete-event HDL simulators together with standard digital circuitry is described. The proposed model can be exploited to specify oscillator requirements or refine them in a system-wide DPLL optimization that may be quickly performed within the same HDL environment. It might be further used to explore and verify system-level performance which would be too complex to handle by an analog simulation. After the target specifications are met, portions of the system's logic can be synthesized from library cells. This eliminates the necessity to use other tools and may dramatically accelerate the design cycle.

Acknowledgements

The authors would like to thank and acknowledge all valuable contributions of the students, faculty, staff and the sponsors of the Berkeley Wireless Research Center.

References

[1] R. B. Staszewski and P. T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*. Wiley, Sep. 2006.

[2] A. Musa, W. Deng, T. Siriburanon, M. Miyahara, K. Okada, and A. Matsuzawa, "A compact, low-power and low-jitter dual-loop injection locked PLL using all-digital PVT calibration," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 50–60, Jan. 2014.

[3] W. Deng, D. Yang, T. Ueno, S. Kondo, K. Okada, and A. Matsuzawa, "A fully synthesizable all-digital PLL with interpolative phase coupled oscillator, current-output DAC, and fine-resolution digital varactor using gated edge injection technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 68–80, Jan. 2015.

[4] G. Marucci, S. Levantino, P. Maffezzoni, and C. Samori, "Analysis and design of low-jitter digital bang-bang phase-locked loops," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 1, pp. 26–36, Jan. 2014.

[5] J. A. Tierno, A. V. Rylyakov, and D. J. Friedman, "A wide power supply range, wide tuning range, all static CMOS all digital pll in 65 nm SOI," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 42–51, Jan. 2008.

[6] M. H. Perrott, "CppSim – a C++ system-level simulator," http://www.cppsim.com, accessed: 2017/10/14.

[7] R. Whittle, "DSP generation of the pink $(1/f)$ noise," www.firstpr.com.au/dsp/pink-noise, accessed: 2011/03/20.