Contributions to the analysis and design of an ADPLL

CYRIL JOUBERT, JEAN FRANÇOIS BERCHER,
GENEVIEVE BAUDOIN
ESIEE / ESYCOM
Noisy le Grand, France
(joubertc, jf.bercher, g.baudoin) @esiee.fr

THIERRY DIVEL, PIERRE BAUDIN

ST-MICROELECTRONICS
Grenoble, France
(thierry.divel, pierre.baudin) @st.com

Abstract— **In this paper, we propose two contributions to the simulation and design of an All-Digital Phase-Locked Loop (ADPLL) for RF applications. First, we extend the behavioral model we already proposed, in order to include detailed fractional aspects. Second, we propose a new adaptive algorithm that can be integrated in this ADPLL in order to lower its hardware complexity, and argue on a recently proposed algorithm for DCO gain estimation. These points are illustrated through simulations.**

I. INTRODUCTION

Staszewski *et al.* recently presented a new All Digital Phase-Locked Loop based RF frequency synthesizer [1].

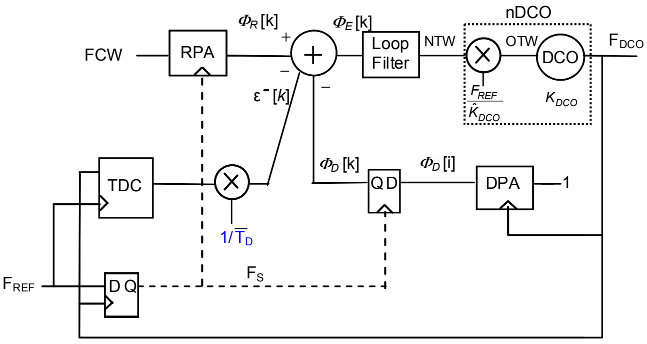

Figure 1. ADPLL based RF frequency synthesizer [1]

A Digitally Controlled Oscillator (DCO) allows for this PLL to be implemented in a fully digital manner [2]. The DCO is normalized using a compensation gain, in such a way that its input, the Normalized Tuning Word (NTW), becomes independent of the gain K_{DCO} of the oscillator.

The Frequency Command Word (FCW) enables to tune the output frequency of the DCO, noted F_{DCO}, according to

$$F_{DCO} = FCW \times F_{REF} \ . \tag{1}$$

Two phase accumulators, the reference phase accumulator (RPA) and the DCO phase accumulator (DPA), are used to count cycle periods of reference and feedback oscillators. A synchronous clock, F_S, undersamples the output of the DPA, so that comparison of the two phases can be performed using the same clock.

The retimed clock, F_S, is achieved by oversampling the reference clock, F_{REF}, by the oscillator clock, F_{DCO}. Note that in Figure 1, index *i* and *k* do not refer to the same clock.

ADPLL precision depends on phase detector performance. One can show that this phase error is proportional to a time delay between input and output clocks.

With phase accumulators, the precision cannot be better than ±1/2 DCO period. Higher ADPLL precision is obtained using fractional phase error correction achieved by Time to Digital Converter (TDC). It is used to convert the delay between DCO and reference clocks directly into a digital quantity [3], with a time resolution, noted Δ*t*, that can be equal to the elementary propagation delay through an inverter gate. This delay is converted to a normalized phase value using normalization factor $1/\overline{T}_D$, where $\overline{T}_D$ is an average value of the DCO period.

In previous work [4] we have proposed a time behavioral model that allows fast simulation (about 100us/s of simulation) and easy access to all variables of interest of this ADPLL. In section II, we present the principle of this model and an improved version taking into account a more precise model of the TDC. In section III, we argue and work out an adaptive algorithm to avoid division operation in phase correction architecture and we argue on a recently proposed algorithm for DCO gain estimation.

II. PRINCIPLE AND MODEL OF PHASE ERROR CORRECTION

The time behavioral model was presented in [4]. In particular, this model recursively computes two important values: τ_k which is the delay between rising edges of reference and DCO clocks, and N[k], an integer value which is the number of DCO periods during a F_{REF} cycle. The index k is linked to F_S clock.

We demonstrated [4] that

$$\begin{aligned}\tau_{k+1} = \tau_k &+ \left(N_i(k) + \tfrac{1}{2}\right)T_D(k) - T_{REF} \\ &- \operatorname{sign}\left(N_i(k)T_D(k) - T_{REF} + \tau_k\right)T_D(k)/2\end{aligned} \tag{2}$$

$$\begin{aligned}N(k) = N_i(k) &+ \tfrac{1}{2} \\ &- \tfrac{1}{2}\operatorname{sign}\left(\tau_k + N_i(k)T_D(k) - T_{REF}\right)\end{aligned} \tag{3}$$

1-4244-0395-2/06/$20.00 ©2006 IEEE.

where $N_i(k) = \lfloor T_D(k)/T_{REF} \rfloor$, $\lfloor \cdot \rfloor$ the rounding down operator and sign(x) is the sign function: sign(x)= –1 if *x<0*, and sign(x)= 1 otherwise.

Then, outputs of phase accumulators can be expressed as

$$\phi_R[k+1] = (\phi_R[k] + FCW) \operatorname{mod}[2^R] \tag{4}$$

and

$$\phi_D[k+1] = (\phi_D[k] + N[k]) \operatorname{mod}[2^D] \tag{5}$$

with R and D, respectively the width of reference and DCO phase accumulators.

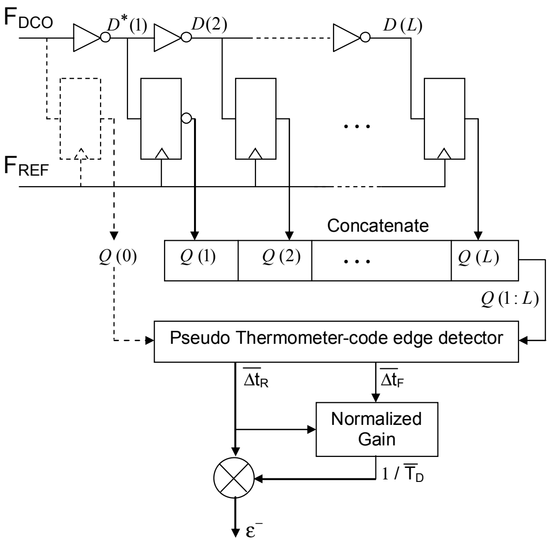

Figure 2. Time-to-Digital Converter (TDC) architecture

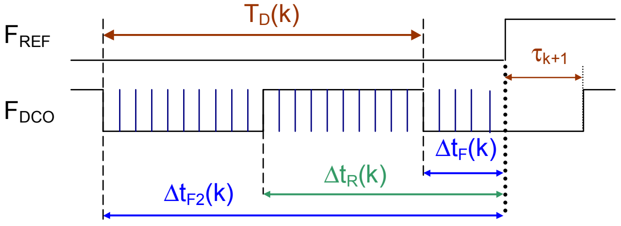

Figure 3. Inputs / outputs of the TDC

In our first approach, the fractional phase error ε was modelized as the quantified version of the delay between rising edges of reference and DCO clocks, normalized to $1/\overline{T}_D$, an averaged value of the DCO frequency.

A more precise model can be derived from the architecture of the TDC. The time-to-digital conversion [3] is realized by passing the DCO signal through a chain of inverters gates of typical delay Δ*t* (as shown in Figure 2). Then, each delayed output is sampled by the same reference clock. The Edge Detector detects first rising and falling edge transitions and use a thermometer to binary encode time differences Δt_R and Δt_F (see Figure 3) into the number $\overline{\Delta t}_R$ and $\overline{\Delta t}_F$ of unit gate delays.

From Figure 3, it is easy to check that $T_D(k) = \Delta t_{F2}(k) - \Delta t_F(k)$, and that the ratio $\beta(k) = |\Delta t_R(k) - \Delta t_F(k)| / T_D(k)$ is the duty cycle $\rho(k)$ if $\Delta t_R(k) > \Delta t_F(k)$ and $1 - \rho(k)$ otherwise.

Using the delay τ_k we can derive outputs of the TDC (with variables defined as on Figure 3)

$$\overline{\Delta t}_R(k) = \begin{cases} \left\lfloor \dfrac{T_D(k) - \tau_{k+1}}{\Delta t} \right\rfloor & if\ \tau_{k+1} \le T_D(k) - \Delta t \\ \left\lfloor \dfrac{2T_D(k) - \tau_{k+1}}{\Delta t} \right\rfloor & \text{otherwise} \end{cases}$$
$$\overline{\Delta t}_F(k) = \begin{cases} \left\lfloor \dfrac{T_D(k)/2 - \tau_{k+1}}{\Delta t} \right\rfloor & if\ \tau_{k+1} \le T_D(k)/2 - \Delta t \\ \left\lfloor \dfrac{3T_D(k)/2 - \tau_{k+1}}{\Delta t} \right\rfloor & \text{otherwise} \end{cases} \tag{6}$$

where $\lfloor \cdot \rfloor$ is the rounding down operator.

With these equations, we take into account all quantization effects. In Particular, we can highlight the incapacity of this architecture to detect a rising edge delay lower than Δt. Indeed, in this case, the output $\overline{\Delta t}_R$ is false and must be corrected. A solution is to add a D flip flop between F_{DCO} and F_{REF} (dashed line in Figure 2). Then, the TDC is able to detect this case and (6) simplifies to:

$$\overline{\Delta t}_R(k) = \left\lceil \frac{T_D(k) - \tau_{k+1}}{\Delta t} \right\rceil$$
$$\overline{\Delta t}_F(k) = \begin{cases} \left\lceil \dfrac{T_D(k)/2 - \tau_{k+1}}{\Delta t} \right\rceil & if\ \tau_{k+1} \le T_D(k)/2 \\ \left\lceil \dfrac{3T_D(k)/2 - \tau_{k+1}}{\Delta t} \right\rceil & \text{otherwise} \end{cases} \tag{7}$$

where $\lceil \cdot \rceil$ is the rounding up operator.

But a new problem appears; $\overline{\Delta t}_R$ is now overestimated and can be, in definite condition, greater than the estimated DCO period. To avoid this situation, we simply subtract 1 to $\overline{\Delta t}_R$ before gain normalization.

Finally, the normalized fractional phase error ε^-, in a fixed-point digital word, can be expressed as

$$\varepsilon^-(k) = \left\lfloor \left(\overline{\Delta t}_R(k) - 1\right) \times \overline{T}_D^{\,-1} \right\rfloor \operatorname{mod}\left[2^F\right] \tag{8}$$

with $F = R - D$ and $\overline{T}_D^{\,-1}$ the inverse average DCO period, used to normalize the TDC output, that is, if the average is evaluated over N_{AVG} DCO periods:

$$\overline{T}_D^{-1} = \left\lfloor \frac{2^F}{\frac{1}{N_{AVG}} \sum_{N_{AVG}} \frac{\left|\overline{\Delta t}_R - \overline{\Delta t}_F\right|}{\beta(k)}} \right\rfloor \tag{9}$$

In practice, it may be possible to suppose that $\rho(k) = 1/2$ (50% duty cycle) in order to reduce the TDC complexity (but of course at the price of a noise term).

A LMS algorithm is proposed in section III in order to compute numerically $\overline{T}_D^{-1}$ without requiring a digital divider.

The phase error is a signed word of width *R*, computed by subtracting 2 unsigned words of same width (ϕ_D and ε^- are concatenated to form a word of width *R*). Thus, equation of phase error can be expressed as

$$\begin{aligned}\phi_E(k) &= \phi_R(k) - (\phi_D(k)2^F + \varepsilon^-(k)) \\ &\quad - 2^R\left(\left\lfloor \frac{\phi_R(k)}{2^{R-1}} \right\rfloor - \left\lfloor \frac{\phi_D(k)2^F + \varepsilon^-(k)}{2^{R-1}} \right\rfloor\right)\end{aligned} \tag{10}$$

where the 2^R correction term permits to take into account that the *R*-width rollovers of inputs are transparent to the phase error (cf. Figure 4).

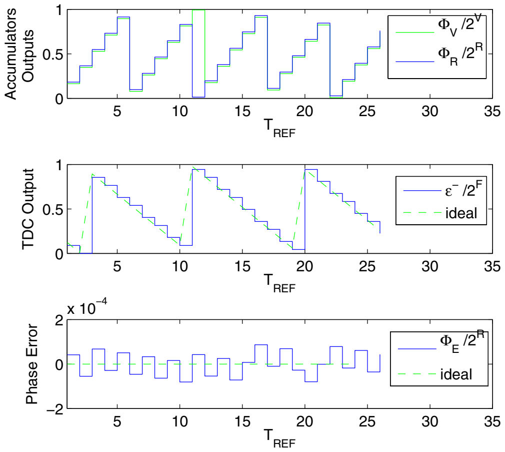

Figure 4. Example of Phase comparison when ADPLL is locked

This extension of our behavioral model enables fast simulations that give identical results to the "circuits" (VHDL) model, while allowing access and control over all variables of interest. These points are of high interest in evaluation and design of new solutions.

III. LMS Algorithms for Gain Estimation

A. LMS Algorithm for Inverse DCO period estimation

The computation of the fractional phase error ε involves the multiplication of the output of the TDC by the inverse period of the DCO, $\overline{T}_D^{-1}$ (cf. Figure 3).

A possible approach [3] is to estimate $\overline{T}_D$ by an average of N_{AVG} values of T_D, and then use a digital divider. $\overline{T}_D$ is then a random value. The inversion can be done via an adaptive method, which results in the inversion without divider, and provides adaptivity to the context.

Let $\alpha = 1/\overline{T}_D$, then clearly, α can be found as a minimizer of $J(\alpha) = \mathrm{E}\left[\left|1-\alpha\overline{T}_D\right|^2\right]$.

Thus, we can adopt the simple LMS algorithm

$$\alpha^{(n+1)} = \alpha^{(n)} - \mu\nabla J \tag{11}$$

with $\nabla J = -2\overline{T}_D\left(1-\alpha\overline{T}_D\right)$, the instantaneous estimate of gradient of the criterion $J(\alpha)$.

A guideline for the choice of the step is $\mu_{opt} = 1/(K.\overline{T}_D^2)$, with K a security factor. Indeed, results on the LMS show that $\mu_{opt} = 1/(K\lambda_{\max})$, where $\lambda_{\max} \approx \overline{T}_D^{\,2}$, the maximum eigenvalue of the correlative matrix. Using this adaptation step and noting that $\alpha \approx 1/\overline{T}_D$ at convergence, we may rewrite the algorithm as

$$\begin{aligned}\alpha^{(n+1)} &= \alpha^{(n)} + \frac{\alpha^{(n)}}{K}\left(1-\alpha^{(n)}\overline{T}_D\right) \\ &= \alpha^{(n)}\left(1+\frac{1}{K}-\frac{\alpha^{(n)}}{K}\overline{T}_D\right)\end{aligned} \tag{12}$$

Complexity can be further reduced using a sign algorithm version:

$$\alpha^{(n+1)} = \alpha^{(n)} + \mu_{opt}\mathrm{sign}\left(1-\alpha^{(n)}\overline{T}_D\right) \tag{13}$$

Note also that instead of using a single estimation of $\overline{T}_D$, and in order to preserve adaptivity, we can use a sliding window, such as

$$\overline{T}_D^{(N)}(k+1) = \overline{T}_D^{(N)}(k) + T_D(k+1) - T_D(k-N) \tag{14}$$

or an exponential mean such as

$$\overline{T}_D(k+1) = \beta\overline{T}_D(k) + (1-\beta)T_D(k+1) \tag{15}$$

with β the forgetting factor.

Simulations of the LMS algorithm for $\overline{T}_D$ inversion using a Matlab and a VHDL implementation are given in Figure 5. These show both effectiveness and fast convergence of the algorithm. Here $\overline{T}_D^{(n)}$ is computed using a sliding window of length 128.

Imprecision from $\overline{\Delta t}_R$ and $\overline{\Delta t}_F$ can be modelized as two uniform random variables, on an interval Δt (the resolution).

Then $T_D(k) = \left|\overline{\Delta t}_R(k) - \overline{\Delta t}_F(k)\right| / \beta(k)$ is distributed according to a triangular distribution, with a variance $\sigma_{T_D}^2 = \Delta t^2 / 6$. Then when the algorithm is iterated at the "sliding rate", (that is each 128 samples of T_D), then the noise is uncorrelated (otherwise, the correlation is simply "the square convolution" of the window shape).

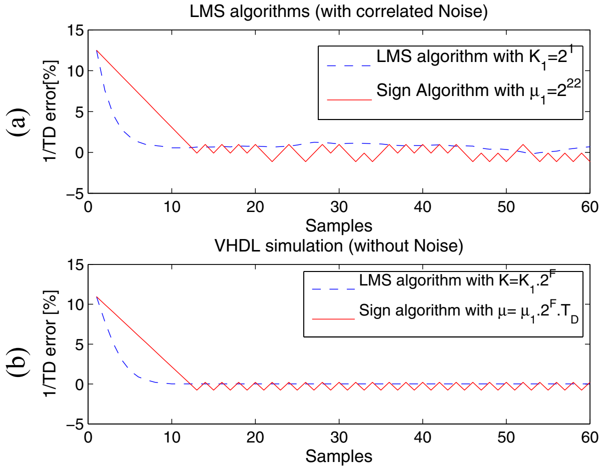

Figure 5. $1/T_D$ error convergence for an initial error of 12%. Matlab (a) and VHDL (b) simulations.

Both algorithms converge easily in less than 15 iterations. Iterating the algorithm at the "sliding rate" reduces consumption and noise but increase the settling time.

Comparison between digital implementation of the LMS algorithm and a digital divider confirm the reduction of the hardware complexity (the integrated circuit area is reduced by ¼ for F=6).

B. *Adaptive DCO Compensation Gain Estimation*

The DCO gain is not precisely known and depends on the operating point. The architecture involves the normalization by an estimated value of the DCO gain.

In a recent publication [5], Staszewski *et al.* propose a LMS adaptation algorithm. In order to estimate the normalization gain $\hat{K}_{DCO}{}^{-1}$, they present the simple adaptation rule:

$$\hat{K}_{DCO}^{-1}[n] = \hat{K}_{DCO}^{-1}[n-1] + \mu\nabla \qquad (16)$$

with a sign algorithm, $\nabla = \phi_E \mathrm{sign}(FCW)$. We show here that such an algorithm can be interpreted as the minimization of a simple criterion.

Indeed, when the PLL is settled, if we apply a ΔFCW step on the Frequency Command Word (FCW), we obtain a deviation of the phase error $\Delta\phi_E = (1-r)\Delta\phi_m$, with $r = K_{DCO} / \hat{K}_{DCO}$, and the phase deviation $\Delta\phi_m$ related to ΔFCW.

Note that this reasoning is only correct for a type I PLL (without filtering effect) or when ΔFCW is applied as a 2 points modulation scheme.

Then, using the minimization criterion $J(r) = \mathrm{E}[|\Delta\phi_E|^2]$, the classical adaptation equation of a gradient algorithm is $r[n] = r[n-1] - \mu\nabla J$ with $\nabla J = 2\,\mathrm{E}[\Delta\phi_E\, \partial\Delta\phi m / \partial r]$ and $\partial\Delta\phi_E / \partial r = -\Delta\phi_m$.

This results in the LMS recursion

$$r[n] = r[n-1] + 2\mu\,\Delta\phi_E\,\Delta\phi_m, \qquad (17)$$

using the instantaneous estimate of ∇J.

Letting $r[n] = K_{DCO}[n] / \hat{K}_{DCO}[n]$, we also obtain

$$F_{REF}K_{DCO}^{-1}[n] = F_{REF}\left(K_{DCO}^{-1}[n-1] + \frac{2\mu}{K_{DCO}}\Delta\phi_E\Delta\phi_m\right) \qquad (18)$$

Last, with $\mu_0 = 2\mu\,F_{REF} / K_{DCO}$ and using the sign of the error, we obtain

$$F_{REF}K_{DCO}^{-1}[n] = F_{REF}\left(K_{DCO}^{-1}[n-1] + \mu_0\Delta\phi_E sign(\Delta\phi_m)\right) \qquad (19)$$

And with $\mathrm{sign}(\Delta\phi_m) = \mathrm{sign}(\Delta FCW)$, we recover the proposed algorithm (16).

IV. Conclusion

In this communication, we have presented an extended behavioral model for simulation and design of an All Digital PLL, including account for fractionnal aspects. The main advantage of such a model is that it enables fast temporal simulations while giving easy access and control to all variables and parameters.

A second contribution of this paper is the presentation or interpretation of adaptive algorithms that can take place in this kind of architecture, in order to lower the hardware complexity.

References

[1] R. B. Staszewski and P. T. Balsara, "Phase-Domain All–Digital Phase–Locked Loop," *IEEE trans. on circuits and System*, vol. 52, no. 3, pp. 159–163, March 2005.

[2] R. B. Staszewski, C–M. Hung, D. Leipold and P. T. Balsara, "A first Multigigahertz Digitally Controlled Oscillator for Wireless Applications," *IEEE Trans. on Microwave Theory and Techniques*, vol. 51, no. 11, pp. 2154–2164, Nov 2003.

[3] R. B. Staszewski, D. Leipold, C–M. Hung and P. T. Balsara, "TDC–Based Frequency Synthesizer for Wireless Applications," *IEEE RFIC Symposium*, pp. 215–218, 2004.

[4] C. Joubert, J. F. Bercher, G. Baudoin *et al.*, "Time Behavorial Model of Phase domain ADPLL based frequency synthesizer," *IEEE Radio Wireless Symposium*, pp 167-170, 2006.

[5] R. B. Staszewski, J. Wallberg, C-M. Hung, G. Feygin, M. Entezari and D. Leipold, "LMS–based Calibration of an RF Digitally–Controlled Oscillator for Mobile Phones, " *IEEE Trans on Circuits and Systems II*, Vol. 53, no. 3, pp 225-229, March 2006