2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools

Behavioural modelling of DLLs for fast simulation and optimisation of jitter and power consumption.

Enrique Barajas, Diego Mateo, José Luis González
Electronic Engineering Department
Universitat Politècnica Catalunya
Barcelona, Spain
Email: ebarajas@eel.upc.edu, mateo@eel.upc.edu, jl.gonzalez@upc.edu

***Abstract*—This paper presents a behavioural model for fast DLL simulations. The behavioural model includes a modelling of the various noise sources in the DLL that produce output jitter. The model is used to obtain the dependence of the output jitter versus the power consumption. The model exploits the open-loop DLL analysis to reduce simulation time when compared to typical DLL evaluation.**

***Index Terms*—DLL, CMOS, behavioural, modelling, Verilog-A, optimisation**

I. Introduction

During the last years the scaling of the CMOS technology has allowed the integration of full systems on a chip (SoC), including both the digital and analog blocks, as well as the RF front-end [1], [2]. However new design difficulties have arose due to this decreasing transistor dimensions [3]–[5]. As it will be demonstrated, once the delay-locked loop (DLL) architecture and size (number of cells) has been fixed, the actual dimensions of the DLL blocks have a great impact on the performance of the system.

The theoretical jitter analysis of all the main contributors to the output jitter has been done for PLL/DLL systems [6], [7]. Also, the transistor level jitter analysis has been carried out for the charge pump [8] and the voltage controlled delay line [9], [10]. This theoretical models have allowed to predict the jitter performance of the DLL blocks and their contribution to the total output jitter. However, their limited accuracy has led to the use of behavioural models based on transistor level simulations, for both PLLs [11] and DLLs [12]. But, even with these latter models, the task to methodically analyse the system for a wide range of dimensions is a very time consuming procedure. This prevents to obtain an accurate model for the DLL jitter performance and the power consumption.

In this paper a new fast behavioural model to analyse the impact of the physical transistor dimensions on the overall performance of a DLL is developed. In section II an introduction to the main sources of jitter in a DLL is carried out, while in section III a behavioural model for the DLL blocks is developed. The basis for fast model DLL simulation are explained in section IV. Finally, the results obtained with the introduced model are analysed and discussed in section V.

II. Jitter analysis of the DLL

The architecture of a delay-locked loop is presented in Fig. 1. The DLL architecture consists of a voltage controlled delay line (VCDL), a phase/frequency detector (P/FD), a charge pump (CP), a loop filter and a regulator. The output of the VCDL can be either used for a clock multiplying DLL or a MUX-based time slot selection. The DLL main sources of jitter are those of the VCDL, the P/FD+CP, the control voltage and the jitter of the reference clock, referred as $\sigma_{_{VL}}$, $\sigma_{_{P/FD+CP}}$, $\sigma_{_{VC}}$ and $\sigma_{_{IN}}$ respectively. Thus the total jitter for the DLL can be expressed as:

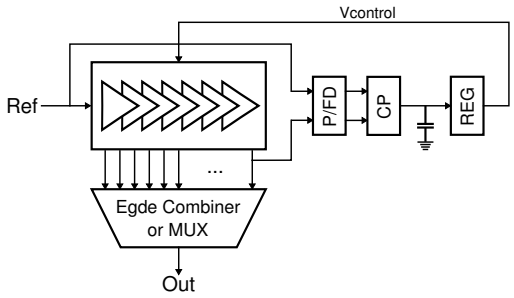

Figure 1. DLL architecture.

$$\sigma^2_{_{DLL}} = \sigma^2_{_{VL}} + \sigma^2_{_{P/FD+CP.}} + \sigma^2_{_{VC}} + \sigma^2_{_{IN}}. \quad (1)$$

In the following subsections these sources of jitter are analysed as a function of the parameters of the DLL, such as the size M, the reference period T_{ref}, loop capacitance, VCDL characteristics, etc.

A. Voltage controlled delay line jitter

Each of the cells that conform the VCDL has its own independent jitter $\sigma_{_{DE}}$. Its contribution to the total added jitter of the VCDL $\sigma_{_{VL}}$ depends on the source of the jitter, and whether it behaves differently when it operates inside the close-loop of a DLL or not. As explained in the appendix, the jitter due to the mismatch of the cells can be compensated at the end of the VCDL, whereas the jitter due to the intrinsic noise of the cells adds along the line. The contributions to the total jitter of the VCDL are, hence, different for the mismatch and noise jitter. The jitter in the m–th cell of a M-size DLL due to mismatch $\sigma_{_{VL_m}}$ and due to noise $\sigma_{_{VL_n}}$

978-0-7695-4171-6/10 $26.00 © 2010 IEEE
DOI 10.1109/DSD.2010.86

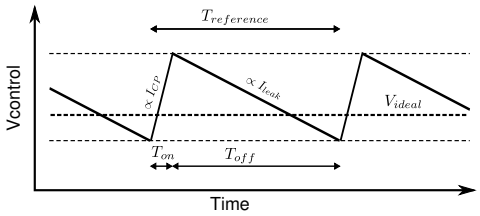

Figure 2. Control voltage ripple.

can be expressed as:

$$\sigma_{VL_m} = \sqrt{\frac{M}{m}(M-m)}\,\sigma_{DE_m} \quad (2)$$

$$\sigma_{VL_n} = \sqrt{m}\,\sigma_{DE_n} \quad (3)$$

where σ_{DE_m} and σ_{DE_n} denote the cell jitter due to mismatch and noise, respectively.

The jitter along the VCDL due to both sources is depicted in Fig. 3 for different number of cells M of the DLL. Note how the jitter due to mismatch is zero at the first and last cells of the VCDL whereas the jitter due to noise is always additive. The jitter scale has been *enhanced* for better readability of the small jitter contributors, but maintaining the jitter order of importance.

B. Control voltage jitter

Another source of jitter in the DLL is the one produced by the variations in the control voltage of the VCDL. This control voltage is responsible for the loop feedback as shown in Fig. 1. The jitter due to the noise of the control voltage is negligible compared to the jitter produced by the control voltage ripple [8], [12], [13]. But even without noise, the control voltage has variations produced by the charge leakage in the loop capacitor.

Assuming as illustrated in Fig. 2 that the leakage current I_{leak} much smaller than the charge pump current I_{CP}, that is $t_{on} \ll t_{off}$, the jitter due to the voltage ripple can be written as:

$$\sigma_{VC} = \tfrac{1}{\sqrt{12}} K_{VL} \frac{T_{ref} I_{leak}}{C_{loop}} \quad (4)$$

where T_{ref} is the period of the reference clock, C_{loop} is the loop capacitance and K_{VL} is the VCDL sensitivity.

There is however another source of frequency variation of the DLL output signal related to the control voltage ripple. Usually, the Jitter is referred to the *variance* of the output signal phase, however there's also a phase drift produced by the *mean*. The physical explanation is that to compensate for the leakage current the DLL loop must produce a slight time mismatch between the reference signal and the output of the last cell, as shown in Fig. 2. This time drift Δ_{VC} can be calculated as:

$$\Delta_{VC} = T_{ref} \frac{I_{leak}}{I_{CP}} \quad (5)$$

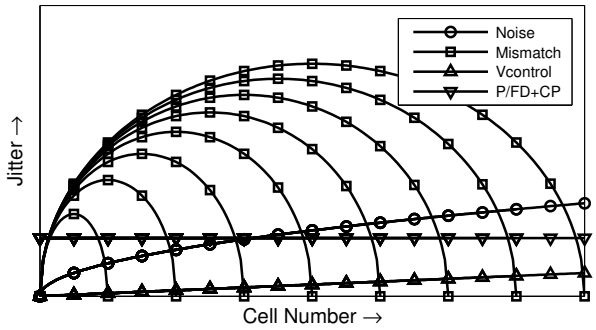

Figure 3. Jitter as a function of the DLL size.

Assuming the loop capacitance is linearly scaled with the size of the VCDL as $C_{loop} = MC_{l0}$, both (4) and (5) can be rewritten as:

$$\sigma_{VC} = \tfrac{1}{\sqrt{12}} K_{DE} \frac{d_0 I_{leak}}{C_{l0}} M \quad (6)$$

$$\Delta_{VC} = d_0 \frac{I_{leak}}{I_{CP}} M \quad (7)$$

where M represents the size of the VCDL, and d_0 and K_{DE} are the nominal delay and sensitivity of the VCDL cell, respectively. As a matter of fact, the frequency offset Δ_{VC} is much lower than the actual jitter σ_{VC} hence is neglected in the jitter calculation.

The jitter due to control voltage ripple, as computed in (6), is represented in Fig. 3 as a function of the DLL size M.

C. Phase/frequency detector and charge pump jitter

The intrinsic noise of the phase/frequency detector and the charge pump can be transformed into an equivalent input jitter. Also the mismatch in the *charge* and *discharge* currents of the charge pump can be translated into an equivalent jitter at the input of the P/FD+CP block [12]. In the same conditions as the control voltage scaling (linear loop capacitance scaling and constant charge pump current) the equivalent jitter in the P/FD+CP due to mismatch —the most predominant source [12]— is independent of the DLL size. This last source of jitter analysed is depicted in Fig. 3.

III. Jitter modelling

To simulate the DLL total jitter a simplified model of the blocks must be developed. These blocks have different sources of jitter but they can all be modelled as a jitter-less block with an equivalent jitter source. However, besides this first order approach, the model needs also to take into account the dimensions of the transistors involved in this blocks. Finally the complete DLL model has to be implemented in a behavioural language (like Verilog-A) to enhance the simulation times.

A. DLL blocks model

As previously analysed, the modelling depends on the jitter source and whether it operates in close-loop or open-loop. The

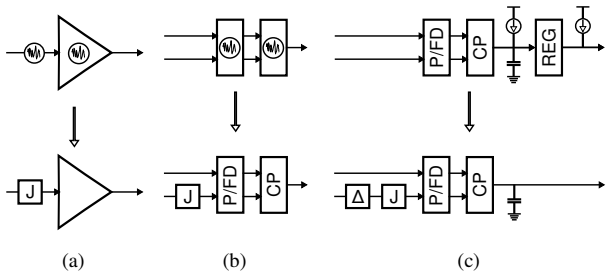

Figure 4. Equivalent jitter modelling for (a) VCDL cell, (b) Phase/Frequency Detector and Charge Pump and (c) loop control voltage ripple.

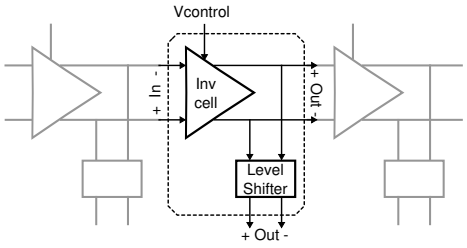

Figure 5. Complementary delay cell in a VCDL.

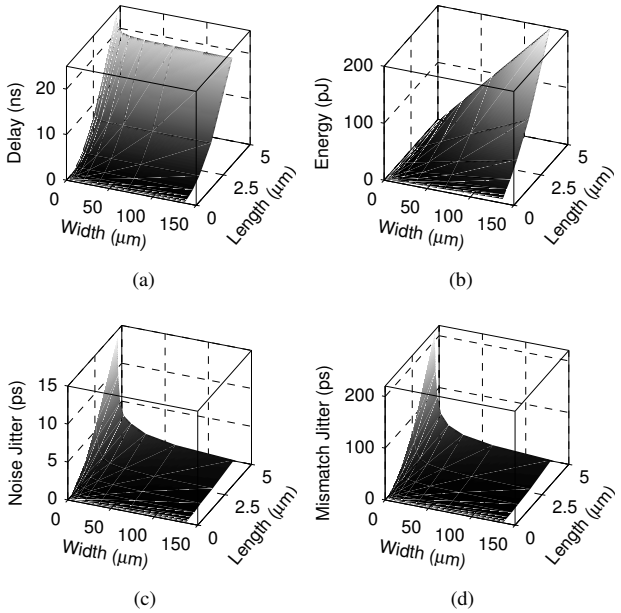

Figure 6. VCDL cell characterisation for W and L sweep. (a) Delay, (b) Energy consumption, (c) Jitter due to noise and (d) jitter due to mismatch.

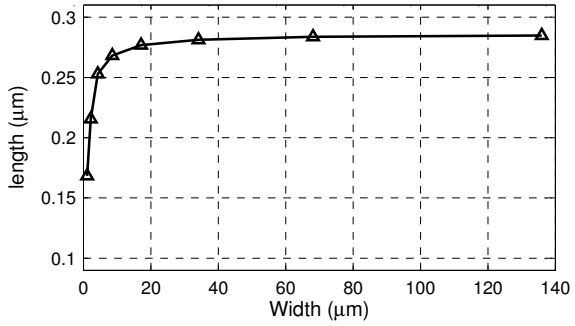

Figure 7. VCDL cell characterisation for constant length and constant delay scaling.

VCDL cell has different sources of jitter as depicted in Fig. 4a; its model is implemented with a jitter-less delay element and an equivalent jitter source, which will be different for the jitter due to mismatch and due to noise. For the phase/frequency detector and charge pump all the noise elements and mismatch sources can be translated as a corespondent jitter source at the input. This modelling is shown in Fig. 4b. Finally the control voltage ripple is modelled as an equivalent jitter and offset source at the input, as represented in Fig. 4c.

B. VCDL block model

The results from the precedent sections shows that the main contributor for the DLL jitter is the VCDL, as depicted in Fig. 3. To study the impact of the transistor scaling in the DLL performance an analysis of the VCDL characteristics relevant to the jitter is developed. The first step is to choose the delay cell architecture and the ratio of the transistors involved. The delay cell examined in this paper is a scaled version of the one presented in [14]. It's a fully differential (complementary) structure whose delay can be controlled by means of the supply voltage of an inverter. To provide rail-to-rail voltage a level shifter is included at the output. The architecture of this cell conforming the VCDL is depicted in Fig. 5.

The next step is to analysed the cell for a combination of different transistor dimensions (width and length). An excerpt of the data populated with this simulation is summarised in Fig. 6.

The modification of the cell dimensions has an impact on the power consumption, the delay, the jitter but also the sensitivity on the control voltage. To be able to modify the cell dimensions without compromising the DLL design the cell delay must be kept constant. The width and length constant-delay space is represented in Fig. 7. The delay has been chosen to be approximately $140\,ps$.

For this constant delay, the jitter can be represented versus the energy of VCDL cell, as shown in Fig. 8. As expected the jitter due to mismatch is an order of magnitude larger than the jitter due to noise. Note, however, that this is the jitter for a single VCDL cell, thus its contribution to the total DLL needs to be adjusted. This problem will be addressed in section IV.

C. Behavioural DLL model

In order to reduce simulation time even further a Verilog-A [15] model of the delay cell used in the VCDL was developed. The code used for modelling the delay cell in Fig. 5 is:

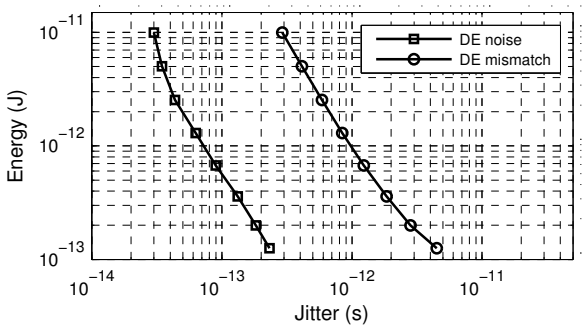

Figure 8. VCDL cell jitter simulation results versus energy consumption.

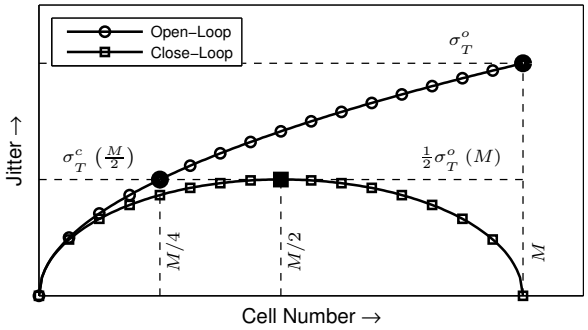

Figure 9. Jitter equivalence for open-loop and close-loop DLL VCDL simulation.

delaycell.va

```
include "disciplines.vams"

module cell_va(VDD,VCNT,VSS,DIP,DIN,DOP,DON,SOP,SON);

output  VDD,VCNT,VSS;
input   DIP,DIN;
output  DOP,DON,SOP,SON;
electrical VDD,VCNT,VSS;
voltage DIP,DIN;
voltage DOP,DON,SOP,SON;

parameter real vtrans     = 0.5; // threshold (V)
parameter real vlogic_high= 1.0;
parameter real vlogic_low = 0.0;
parameter real tdel    = 1p from [0:inf];
parameter real trise   = 1p from [0:inf];
parameter real tfall   = 1p from [0:inf];
parameter real tjitter = 0p from [0:inf];
parameter real iVDD    = 0  from [0:inf];
parameter real iVCNT   = 0  from [0:inf];
parameter integer initseedN = -500;
parameter integer initseedP = -700;
real vop,von;
integer seedn, seedp;

analog begin

  V(DOP)<+ transition(vop,tdel+tjitter*$rdist_normal(seedp,0,1),trise);
  V(SOP)<+ transition(vop,tdel+tjitter*$rdist_normal(seedp,0,1),trise);
  V(DON)<+ transition(von,tdel+tjitter*$rdist_normal(seedn,0,1),tfall);
  V(SON)<+ transition(von,tdel+tjitter*$rdist_normal(seedn,0,1),tfall);
  I(VDD,VSS) <+ iVDD;
  I(VCNT,VSS)<+ iVCNT;

  @(initial_step) begin
    seedn=initseedn;
    seedp=initseedp;
  end
  @(cross(V(DIP) - vtrans, +1)) begin
    vop=vlogic_high;
    von=vlogic_low;
  end
  @(cross(V(DIP) - vtrans, -1)) begin
    vop=vlogic_low;
    von=vlogic_high;
  end
 end
endmodule
```

It includes delay and jitter modelling as well as the power consumption. The data is obtained from the look-up table of the simulation results of a single delay cell as shown in Fig. 6.

IV. Open-loop DLL model

In the previous sections a behavioural model for the DLL blocks providing energy, jitter and delay data was developed. However, the DLL simulation still needs an extremely long transient to lock the DLL loop. In this section a novel approx to overcome this problem is discussed.

The expressions for the close-loop and open-loop jitter obtained in the appendix can be written as:

$$\sigma_{_T} = \sqrt{\frac{M}{m}(M-m)}\,\sigma_{_E} \tag{8}$$

$$\sigma_{_T} = \sqrt{m}\,\sigma_{_E} \tag{9}$$

Denoting the close-loop jitter as $\sigma^c_{_T}$ and the open-loop jitter as $\sigma^o_{_T}$, they can be related as:

$$\sigma^c_{_T}\left(m^c\right) = \sigma^o_{_T}\left(m^o\right) \tag{10}$$

From (8) and (9):

$$\sqrt{\frac{M}{m}(M-m)}\,\sigma^c_{_E} = \sqrt{m}\,\sigma^o_{_E} \tag{11}$$

Assuming $\sigma^c_{_E} = \sigma^o_{_E}$ and for the close-loop worst-case jitter scenario ($m = \frac{M}{2}$):

$$\sigma^c_{_T}\left(\tfrac{M}{2}\right) = \sigma^o_{_T}\left(\tfrac{M}{4}\right) \tag{12}$$

Consequently the equivalent close-loop jitter for $\frac{M}{2}$ can be calculated as the open-loop jitter for $\frac{M}{4}$. Furthermore, the open-loop jitter for M can be estimated by simulation and then, with the expression in (9), the close-loop jitter can be obtained as:

$$\sigma^c_{_T}\left(\tfrac{M}{2}\right) = \tfrac{1}{2}\sigma^o_{_T}\left(M\right) \tag{13}$$

These two relations are depicted in Fig. 9 for a fixed DLL size. The jitter in open-loop and its equivalent in close-loop from (12) are shown. It's also shown the open-loop equivalence relation between the jitter of the *quarter* and the last cell of the VCDL in (13).

As represented in Fig. 10, the DLL can be simulated in open-loop (reducing it to only the VCDL) and then equalise the jitter whenever appropriate. Thus the jitter due to noise in the VCDL, the jitter due to the control voltage ripple and the P/FD+CP jitter is simulated directly. The jitter due to mismatch in the VCDL must be corrected using (13), though.

The relation in (13) is used to greatly speed-up the DLL simulation: Since the period T_{ref} is known, the *long* transient simulation to the steady-state close-loop analysis can be skipped. This is specially important in montecarlo simulations where this loop-locking transient simulation cannot be reused between runs.

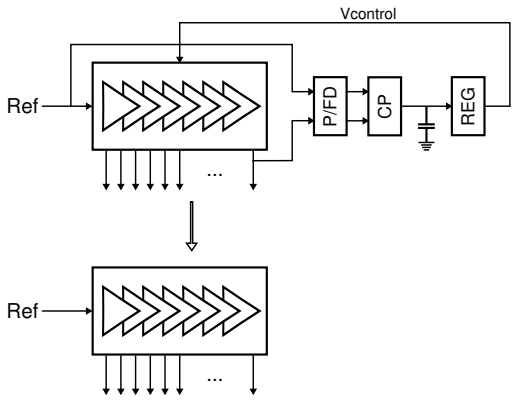

Figure 10. Open-loop equivalent DLL model.

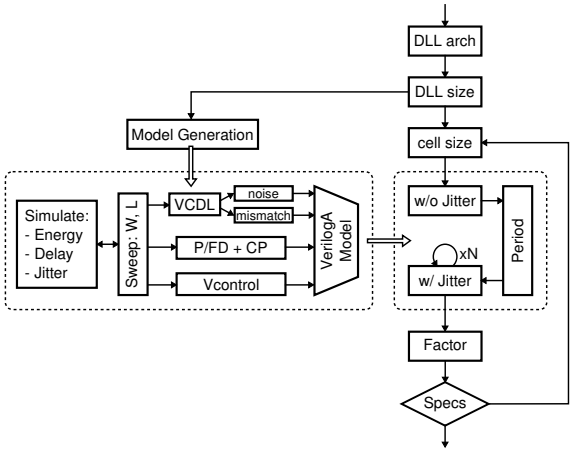

Figure 11. Model generation and simulation algorithm.

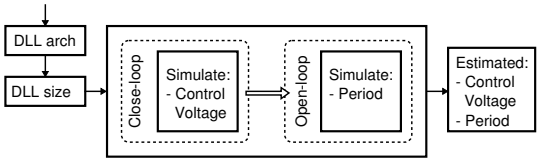

Figure 12. Estimation of control voltage and period for open-loop operation of the DLL.

V. Simulation algorithm and results

A. Simulation algorithm

The algorithm required to simulate the DLL under the conditions described in this papers is depicted in Fig. 11. Once the DLL architecture has been set, including all the components topology and size, a Verilog-A model must be developed. The phase/frequency detector with charge pump jitter and control voltage ripple are estimated by simulation, together with the jitter due to mismatch and due to noise in the VCDL. Other characteristics such as energy consumption and the delay are also estimated by simulation. This model development is swept over the VCDL cell physical dimensions to generate a multidimensional model in Verilog-A.

In the DLL close-loop operation, the loop sets the control voltage to match the DLL period to that of the reference input signal. However, as explained in section IV the behavioural DLL model is simulated in open-loop. Hence, the first step is to simulate the DLL in close-loop operation and obtain the control voltage in the steady state. Then, assuming the control voltage constant to this simulated value, and without any source of jitter, the DLL period T_{ref} is estimated by simulation. This later simulation is performed in open-loop operation, thus it only needs one period to get to the steady state. Although the period and control voltage found with this open-loop method is slightly different to the close-loop operation, the differences in both power consumption and jitter are negligible, whereas the simulation speed in much faster. This procedure is depicted in Fig. 12.

With the period acquired in the previous step, the final simulation with jitter is carried out. This final simulation is iterated N times in order to obtain a statistically correct estimation of the jitter. Depending on the jitter source, a correction factor must be applied, as explained in section IV. Finally a new cell size must be chosen if the simulated jitter doesn't fit into the specifications. Or alternatively the cell size can be swept over the constant-delay space defined in Fig. 7 to simulate the energy versus jitter interdependence.

The simulation algorithm was developed in the MATLAB environment [16]. The transistor level and Verilog-A simulations were performed with the Spectre RF simulator [17].

B. Simulation results

The Verilog-A model described in the previous section was developed for a DLL implemented in a $90\,nm$ CMOS technology. The reference frequency of the input signal was set to $200\,MHz$. The DLL was fixed to 36 VCDL cells, hence the nominal delay of the VCDL cell was determined to be around $140\,ps$. The simulation results for the behavioural model described are presented in Fig. 13.

As expected, the most predominant source of jitter in the DLL is the mismatch jitter in the VCDL. Its dependence on the energy is consistent with the results presented in [10]. Although simulations are not accurate for small values of the jitter due to noise in the VCDL because of the numeric rounding, this jitter is one order of magnitude lower than the jitter due to mismatch and thus can be safely ignored. On the contrary, the jitter due to the phase/frequency detector and charge pump and due to the control voltage ripple are not scaled with the VCDL cell dimensions, hence are not always negligible.

For the jitter due to mismatch, the behavioural model simulation results match those of the transistor level simulation, as shown in Fig. 13. These results confirm the accuracy of the methodology presented in this paper. On the other hand, the jitter due to noise in the behavioural model has an slight error for very low jitter results when compared to transistor level simulations. However, as depicted in Fig. 13, the jitter due to mismatch is the predominant source of jitter and thus the jitter

Table I
SIMULATION TIME COMPARATIVE

Simulation	time [a]		speed-up
	Behavioural open-loop	CMOS close-loop	
w/o jitter	$1.3\,ks$	$470\,ks$	361
w/ jitter	$57\,ks$	$21\,Ms$	368

[a] equivalent single threaded on a Intel® Xeon® CPU E5520 @ 2.27GHz

due to noise can be neglected. The jitter due to control voltage ripple and phase/frequency detector was obtained directly from transistor level simulations, thus the results for the behavioural model match perfectly. Note that due to the fact that transistor level simulations are extremely long, fewer sweep point have been simulated.

The simulation time for this behavioural modelling of the DLL is much lower than a CMOS close-loop simulation. This time includes the setup simulation to obtain the period, the N-runs simulation to obtain the jitter and also the delay cell modelling, as summarised in Table I. The results show that the new open-loop behavioural model introduced in this paper is 360 times faster than a full close-loop CMOS transistor level analysis. Both the close-loop transistor level and the open-loop Verilog-A simulations can be greatly parallelled, therefore reducing drastically the total simulation time. Although the simulations were in fact parallels, the time in Table I is the added equivalent time of all these simulations.

VI. CONCLUSION

A new behavioural model for fast simulation of DLLs has been presented in this paper. This procedure allows to easily simulate the DLL jitter performance for various VCDL cell dimensions, at a fraction of the time needed in normal DLL analysis. Hence, the main contributor to the DLL jitter can be independently analysed, and the DLL power consumption performance versus the jitter can be obtained.

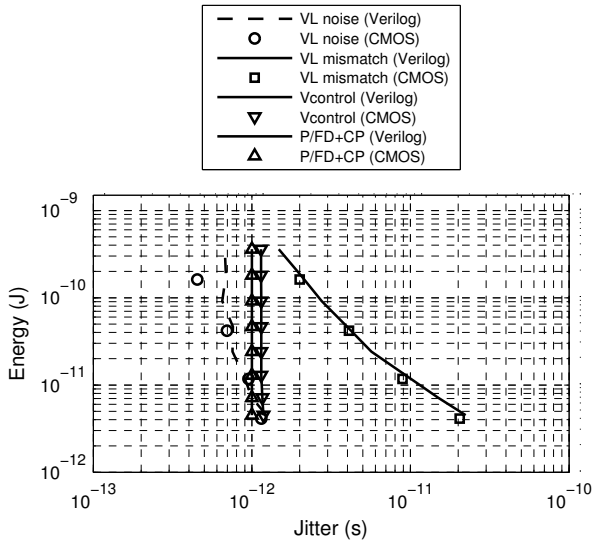

Figure 13. DLL jitter simulation results versus energy consumption.

APPENDIX
OPEN AND CLOSE-LOOP JITTER ANALYSIS OF A VCDL

In this section the total jitter of a VCDL is evaluated. In an open-loop analysis of the VCDL, where all the cells are uncorrelated, the jitter along the line increases monotonically and thus the total jitter is unbounded. Nonetheless, in a delay-locked loop they behave much differently. The DLL phase/frequency detector compares the edges of the first and last cells' output of the VCDL; the stationary delay errors like those due to mismatch can be therefore compensated for the first and last cells. On the other hand, the noise produces fast-varying delay errors (which can't be compensated) that yield to an *effectively* open-loop operation. Hence, the total jitter of a VCDL operating in a DLL can be calculated, in a first-order approximation, as a combination of the jitter *inside* the loop bandwidth and *outside* it.

In the analysis of a DLL, the mismatch jitter must be modelled as a close-loop jitter, whereas the noise jitter must be modelled as an open-loop jitter. The following subsections analyse the jitter along the VCDL in these two cases.

A. Close-loop

Let d_0 be the nominal jitter-less delay of the cell and ξ_n the error of the n–th cell; thus total delay the n–th cell will be:

$$d_n = (1 + \xi_n)\, d_0 \tag{14}$$

Thus the jitter of a VCDL cell can be expressed as:

$$\sigma_{_{E_n}} = E\{\xi_n\}\, d_0 = \sigma_{\xi_n} d_0 \tag{15}$$

The cell jitter can be assumed to be uncorrelated between cells and constant, therefore:

$$\sigma_{_E} = \sigma_\xi d_0 \tag{16}$$

For a voltage controlled delay cell implemented with M cells, the period T_{ref} in close-loop is:

$$T_{ref} = \sum_{n=1}^{M} d_n \tag{17}$$

An expression for the jitter from (14) and (17) can be derived as explained in [10]:

$$\sigma^2_{\Delta t_m} = \frac{T^2_{ref}}{M^3} m(M-m)\sigma^2_\xi \tag{18}$$

Or as an expression of the cell jitter as:

$$\sigma_{_T} = \sqrt{\frac{M}{m}(M-m)}\,\sigma_{_E} \tag{19}$$

For the initial $m = 0$ and last cell $m = M$ the jitter in close-loop is zero, as shown in Fig. 9 for a fixed DLL size M. The maximum jitter is obtained for $m = \frac{M}{2}$.

B. Open-loop

In this case the total delay error is unbounded due to the jitter, therefore (17) is transformed into:

$$T_{ref} \neq \sum_{n=1}^{M} d_n \tag{20}$$

The m–the cell delay error the VCDL can be expressed as:

$$\begin{aligned} \Delta t_m &= \sum_{n=1}^{m} d_n - \frac{m}{M} T_{ref} \\ &= T_{ref} \left(\frac{m + \sum_{n=1}^{m} \xi_n}{M} - \frac{m}{M} \right) \\ &= \frac{T_{ref}}{M} \sum_{n=1}^{m} \xi_n \end{aligned} \tag{21}$$

Hence the variance can be calculated as:

$$\sigma^2_{\Delta t_m} = E\left\{ {\Delta t_m}^2 \right\} = E\left\{ \frac{T^2_{ref}}{M^2} \left(\sum_{n=1}^{m} \xi_n \right)^2 \right\} \tag{22}$$

Thus finally,

$$\sigma^2_{\Delta t_m} = \frac{T^2_{ref}}{M^2} m\, \sigma^2_{\xi} \tag{23}$$

Or alternatively as an expression of the cell jitter as:

$$\sigma_{_T} = \sqrt{m}\, \sigma_{_E} \tag{24}$$

This equation can also be derived from [9]:

$$\sigma^2_{\Delta t_M} = \sigma^2_{\xi_n} M \frac{2}{2 - \varepsilon} \tag{25}$$

assuming the *loop gain* ε equal to zero.

As represented in Fig. 9, the jitter increases monotonically with the cell number m (up to the DLL size M).

ACKNOWLEDGEMENTS

This work has been partially supported by EU-FEDER funds, TEC2008-01856 project and AGAUR SGR 1497 funds.

REFERENCES

[1] H. Xie, S. Fan, X. Wang, A. Wang, Z. Wang, and H. Chen, "A Pulse-Based Full-Band UWB Transceiver SoC in 0.18 um SiGe BiCMOS," in *SOC Conference, 2006 IEEE International*, sept. 2006, pp. 73–76.

[2] P. van Zeijl, "Wireless transceiver design for SoC and SDR," in *Radio and Wireless Symposium, 2008 IEEE*, jan. 2008, pp. 807–810.

[3] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *Solid-State Circuits, IEEE Journal of*, vol. 24, no. 5, pp. 1433–1439, oct 1989.

[4] B. Davari, R. Dennard, and G. Shahidi, "CMOS scaling for high performance and low power-the next ten years," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 595–606, apr 1995.

[5] K. Lee, I. Nam, I. Kwon, J. Gil, K. Han, S. Park, and B.-I. Seo, "The impact of semiconductor technology scaling on CMOS RF and digital circuits for wireless application," *Electron Devices, IEEE Transactions on*, vol. 52, no. 7, pp. 1415–1422, july 2005.

[6] B. Kim, T. Weigandt, and P. Gray, "PLL/DLL system noise analysis for low jitter clock synthesizer design," in *Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on*, vol. 4, 30 1994, pp. 31 –34 vol.4.

[7] N. Godambe and C.-J. Shi, "Behavioral level noise modeling and jitter simulation of phase-locked loops with faults using VHDL-AMS," in *VLSI Test Symposium, 1997., 15th IEEE*, apr-1 may 1997, pp. 177–182.

[8] R. Aguiav and D. Santos, "Modelling charge-pump delay locked loops," in *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on*, vol. 2, sep 1999, pp. 823–826 vol.2.

[9] R. van de Beek, E. Klumperink, C. Vaucher, and B. Nauta, "Analysis of Random Jitter in a Clock Multiplying DLL Architecture," in *Proceedings of ProRisc 2001*. Utrecht, The Netherlands: STW, Technology Foundation, 2001. [Online]. Available: http://doc.utwente.nl/42541/

[10] ——, "A Jitter in DLL-Based Clock Multipliers caused by Delay Cell Mismatch," in *Proceedings of ProRisc 2002*, Veldhoven, 2002. [Online]. Available: http://doc.utwente.nl/43757/

[11] M. Sayfullah, B. Roland, and A. Scholtz, "Jitter analysis of a mixed PLL-DLL architecture," in *Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on*, dec. 2008, pp. 750–754.

[12] E. Barajas, R. Cosculluela, D. Coutinho, D. Mateo, J. Gonzalez, I. Cairo, S. Banda, and M. Ikeda, "Behavioral Modeling of Delay-Locked Loops and its Application to Jitter Optimization in Ultra Wide-Band Impulse Radio Systems," in *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, april 2007, pp. 1–6.

[13] M. Pons, F. Martorell, X. Aragones, F. Moll, and A. Rubio, "Ground bounce modelling for digital gigascale integrated circuits," in *International Conference on Design and Test of Integrated Systems in Nanoscale Technology, DTIS*, 2006, pp. 305–309.

[14] E. Barajas, D. Gomez, D. Mateo, and J. Gonzalez, "A 75 pJ/bit All-Digital Quadrature Coherent IR-UWB Transceiver in 0.18 μm CMOS," in *Radio Frequency Integrated Circuits Symposium, 2010. RFIC 2010. IEEE*, may 2010.

[15] K. Kundert and O. Zinke, *The Designer's Guide to Verilog-AMS*, 2004.

[16] MATLAB version 7.7.0, 2008, *The Mathworks Inc.*, Natick, Massachusetts. [Online]. Available: www.mathworks.com

[17] MMSIM Spectre RF Simulator version 6.2.0, 2007, *Cadence Design System Inc.*, San Jose, California. [Online]. Available: www.cadence.com