

Event Driven Analog Modeling for the Verification of PLL Frequency Synthesizers

Yifan Wang
Chair of Integrated
Analog Circuits
Sommerfeldstrasse 24
RWTH Aachen,
Germany
wang@ias.rwth-
aachen.de

Christoph
Van-Meersbergen
Chair of Integrated
Analog Circuits
RWTH Aachen,
Germany
mailbox@ias.rwth-
aachen.de

Hans-Werner Groh
Atmel Automotive GmbH
Theresienstrasse 2
74072 Heilbronn,
Germany
Hans-Werner.Groh
@atmel.com

Stefan Heinen
Chair of Integrated
Analog Circuits
Sommerfeldstrasse 24
RWTH Aachen,
Germany
mailbox@ias.rwth-
aachen.de

ABSTRACT

The focus of this work is to provide a efficient modeling approach for the functional verification of complex analog frequency synthesizers. The event driven analog modeling approach uses the double precision data type wreal (supported by VerilogAMS), that enables analog accuracy in the digital simulation domain. It is therefore possible to separate high frequency signal paths in the frequency synthesizers from the analog domain, in order to archive higher simulation efficiency for fast verification purposes. The modeling approach and an investigation of the necessary accuracy requirements for the verification including phase noise performances of the analog frequency synthesizers is presented. The proposed approach is demonstrated on baseof a sub micron CMOS Fractional-N frequency synthesizer and compared with different traditional modeling approaches, like phase model and pure digital model. The paper concludes with a proposal of a verification approach for RF mixed signal systems.

General Terms

Modeling, VerilogAMS, wreal, verification, phase noise, accuracy requirements

1. INTRODUCTION

Verification is becoming a key task in today's design of complex, highly integrated circuits [1]. More and more digital functionality enters the formerly analog-only radio frequency (RF) circuit blocks. Since the digital calibrated analog building blocks need a huge number of digital connections and programming routines, a verification of the complex interaction between all of them is inevitable. The mission of so called functional verification prior to tape-out is not only checking the correctness of wire connections and signal routing, but also making a rough performance analysis "over night" - already during the design process. The

target of this work is to provide a modeling and simulation strategy that tries to fulfill this requirement.

Modeling is a indispensable task for the verification of a complex mixed-signal system. A transient system simulation on transistor level of the analyzed fractional-N PLL (Phase Locked Loop) schematics would take days or even weeks to complete. Using the proposed modeling approach we can reduce this time to approx. 1 hour. A lot of information is lost at the cost of this speedup, but it is still sufficient to provide a functionality and connectivity test including the analysis of the overall PLL performance like settling time, spurious and phase noise.

In the following an brief introduction to analog and digital domain simulations, including supported data types and their advantages as well as event driven modeling is given. The new approach of event driven analog modeling for the verification of frequency Synthesizers is presented in the section 4 and 5. Section 5 demonstrates the approach using a commercially available fractional-N analog frequency synthesizer shown in figure 1.

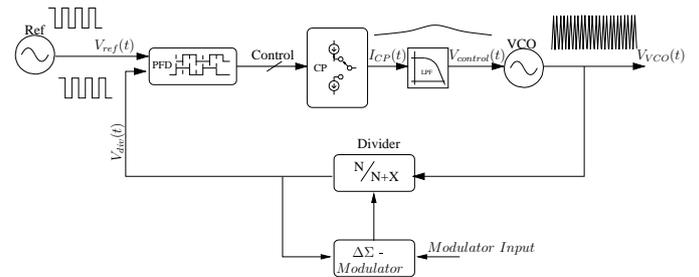


Figure 1: Block level structure of the investigated fractional-N PLL frequency synthesizer.

2. SIMULATION DOMAINS

Current simulators offer two simulation domains: analog and digital. In each of those domains specific restrictions apply. The question is - which domain to choose for a specific building block? It is therefore important at this point not to get confused and to distinguish between signal values and simulation domains. While signals in the analog domain are continuous time and continuous value, signals in the digital

domain are discrete time and (depending on the type) continuous value or digital value. For the analog domain the simulator generates a matrix of differential equations, based on Kirchhoff's flow and potential laws, which are solved by the Newton-Raphson-Method. Any signal change leads to slightly different elements in the node admittance matrix, which has to be solved in an iterative fashion to meet the desired convergence criteria (accuracy). Advanced simulation methods optimized for the high frequency analog portion, such as Periodic Steady State (PSS) or Harmonic Balance (HB), are always taking special requirements on the test case into account, so that they are not suitable for a mixed-level or mixed-domain simulation, which especially integrates digital algorithms into analog signals. The digital domain simulation uses an event-driven (also known as data-flow) approach, which leads to a sensitivity list, where each signal change triggers new calculations at the connected nodes, but not for the whole system as the analog simulator. It is obvious, that the digital domain uses the aspect of time (data-flow : one calculation after the other) and therefore does only provide transient simulation possibilities. For mixed-signal systems, which consist of analog as well as digital elements, both solvers are necessary - each one to simulate it's own portion of the system and continuously synchronizing with the other domain. One important aspect here is to mention, that digital solvers are not capable of iterating backward in time to our knowledge. Digital concepts therefore can't handle iterative features as used in backward euler or trapezoidal formulas. Since the signal of the $\Delta\Sigma$ Modulator in s fractional-N PLL system is random and it therefore doesn't provide the basic, periodic conditions for the PSS analysis [2]. The only simulation method that is currently implemented in an acceptable fashion for the verification of such kind of mixed-signal circuits is still transient.

3. EVENT-DRIVEN ANALOG MODELING

Event-driven analog modeling [3] leads to an approach excluding the high frequency signal path from the analog domain, e.g. using the wreal data type introduced by Verilog-AMS. To our knowledge, approaches to partition the analog signals into loosely coupled sub-matrices have not been very successful up to now. Using the wreal data type the high frequency signal path can be extracted from the analog matrix and put into the digital, event-driven domain, while keeping analog behavioral models or circuits for low frequency like biasing concepts (e.g. current mirrors), bandgaps and analog filters. In the event driven approach, the size of the equation systems that has to be solved for each of these time steps will be reduced dramatically, since only isolated portions are calculated but not the whole analog matrix. This leads to a much shorter simulation time, but also to the necessity to map every block-specification into time-domain, which is difficult for specifications that require precise frequency domain data.

4. PROPOSED BEHAVIORAL MODELS

Consider the different modeling contributions for PLL frequency synthesizers, different approaches has been presented in the past: 1.)The phase domain approach, which has been presented in [4] and [5], uses linear phase domain models to predict the influence of phase noise. In this case, the Voltage Controlled Oscillator (VCO) model is the most time

consuming block in the PLL, since it was implemented in the analog domain. In [4] and [6], the model of divider has been merged into the VCO model in order to increase the simulation efficiency. While this modeling approach is well suited for the start of design circle to estimate the overall noise of the PLL system, it has some drawbacks when it comes to the verification for given design. On the one hand the realization of merged VCO+divider doesn't correspond to the block level circuit structure, which makes the verification of connectivities and debugging of single blocks on transistor level almost impossible. One the other hand, using analog statements like "@cross()" or "transition" in the VCO model is not efficient enough as we will see late in section (4.2). 2.)The event driven approach [7] is originally developed for pure digital circuits. It formulates the phase noise into its equivalent jitter in the time domain. Compare to the phase-domain models, the digital models have higher simulation efficiency, but it is also not well suited for the verification of analog frequency synthesis due to the pure digital implementation. Our approach combines the advantages of the aforementioned methods. On the one hand it is possible to model each of the blocks in the event driven domain in order to increase the simulation efficiency. On the other hand the models can be embedded into test benches on transistor level for a mixed-level simulation, since the wreal data type enables analog precision in the event driven domain. However, special calculation has to be done in order to accurately map the block-specification from frequency-into time-domain while keeping the simulation efficiency as high as possible, as we will see in this section.

4.1 Nonuniform sampled loop filter

One consideration should be made when using the event-driven approach with frequency domain filters: While continuous time calculations are obviously not applicable, one might argue to go for standard discrete time filter methods. Sadly, these would require fixed sampling time instances, so that the main benefit of the event-driven approach would be lost [3]. Since the time in simulation of the discrete time kernel can only be increased (there's no iterative possibility like in the analog domain), the only possible method to do a discrete time filtering is using a Forward Euler approximation as shown in [8]. A first order filter having a transfer function of:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b_0 + b_1 s}{a_0 + a_1 s}, \quad (1)$$

can be transferred into the time domain as:

$$a_0 y(t) + a_1 \dot{y}(t) = b_0 x(t) + b_1 \dot{x}(t). \quad (2)$$

Using the mentioned Forward Euler approximations:

$$x(t) = x(t - \Delta t) + \dot{x}(t - \Delta t) \cdot \Delta t \quad (3)$$

and

$$y(t) = y(t - \Delta t) + \dot{y}(t - \Delta t) \cdot \Delta t \quad (4)$$

leads to:

$$y(t) = \frac{a_0 \cdot x(t - \Delta t) + a_1 \cdot \dot{x}(t - \Delta t) - b_1 \cdot \dot{y}(t - \Delta t)}{b_0 - \frac{b_1}{\Delta t}}. \quad (5)$$

Identical results can be calculated for a 2nd order filter as:

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b_0 + b_1 \cdot s + b_2 \cdot s^2}{a_0 + a_1 \cdot s + a_2 \cdot s^2}. \quad (6)$$

Displayed in time domain, following differential equation is obtained:

$$a_0 y(t) + a_1 \dot{y}(t) + a_2 \ddot{y}(t) = b_0 x(t) + b_1 \dot{x}(t) + b_2 \ddot{x}(t). \quad (7)$$

Substituting $y(t) = \dot{u}(t)$, $x(t) = \dot{v}(t)$ and integrating on both sides leads to:

$$a_0 u(t) + a_1 y(t) + a_2 \dot{y}(t) = b_0 v(t) + b_1 x(t) + b_2 \dot{x}(t). \quad (8)$$

With these auxiliary variables u, v an equivalent filter using a Forward Euler approximation can be realized as:

$$y_n = \frac{\dot{x}_n h_n b_2 + h_n x_n b_1 + h_n v_n b_0 + y_{n-1} a_2 - h_n u_n a_0}{h_n a_1 + a_2}. \quad (9)$$

Higher order filters can be implemented as cascade of 1st and 2nd order filters. The circuit representation of the loop filter is shown in figure 2. A part of the nonuniform sampled filter model is shown in listing 1:

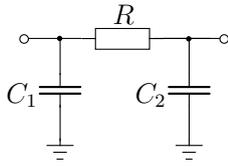


Figure 2: Analog loop filter structure.

```

module LF (out, in);
output out; input in;
wreal out, in;
parameter real r, c1, c2;
real hn, hn1, vn, vn1, un, un1, xn, xn1;
real yn, yn1, dxn, dxn1, dyn1, act_time, prev_time;

initial begin
.. initialize the coefficients ...
end
assign out=yn;

always@(in) begin
xn=in;
act_time=$realtime;
hn=1.0 f*(act_time-prev_time);
dxn=(xn-xn1)/hn;
vn=vn1+hn*xn1;
un=un1+hn*yn1;
yn=((z0*vn+z1*xn+z2*dxn-p0*un)/p1
+ p2/p1*yn1/hn)/(1.0+p2/(p1*hn));
prev_time=act_time;
dyn1<=(yn-yn1)/hn;
dxn1<=dxn; xn1<=xn;
yn1<=yn; un1<=un;
vn1<=vn; hn1<=hn;
end
endmodule

```

Listing 1: Example source code for the 2nd order nonuniform sampled loop filter.

The comparison between the filter characteristic of the nonuniform sampled loop filter and its analog implementation is presented in figure 3. The model implementation maps the frequency domain specification from the loop filter sufficiently for verification purpose. The simulation time for 100 μ s transient analysis on the other hand, has been reduced by a factor of five.

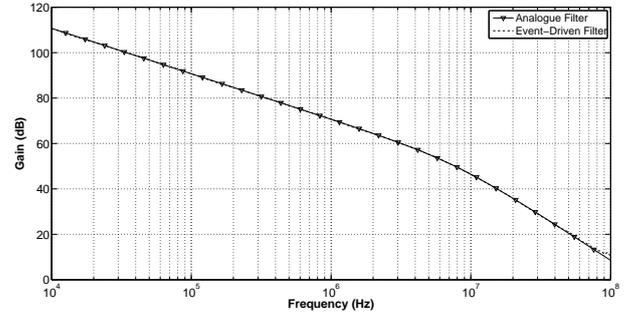


Figure 3: Comparison of the AC characteristic of the analog and the nonuniform sampled loop filter.

4.2 Voltage Controlled Oscillator

The event driven analog VCO model structure is presented in figure 4. First the input voltage signal is converted to the output frequency or rather in the desired oscillation period. Then the timing jitter is computed and added to the period, which leads to the noisy VCO output signal. Our target is to provide a VCO model with high simulation efficiency, which also maps the given specification from circuit level accurately enough in order to fulfill the aforementioned verification requirements.

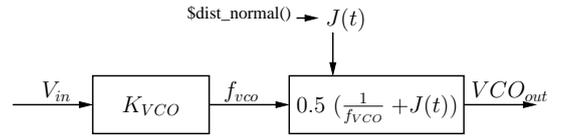


Figure 4: Wreal VCO model structure.

According to [9] and [6], the variance σ for a timing jitter can be calculated from phase noise measurements based on following equations:

$$c = \frac{10^{\frac{PNdBc}{10}} \cdot \Delta f^2}{f_0^2} \quad (10)$$

$$J = \sigma = \sqrt{\frac{c}{f_0}} = \sqrt{\frac{10^{\frac{PNdBc}{10}} \cdot \Delta f^2}{f_0^3}} \quad (11)$$

This can be proven as shown in [6] and by transient simulation result from the wreal VCO model in figure 5.

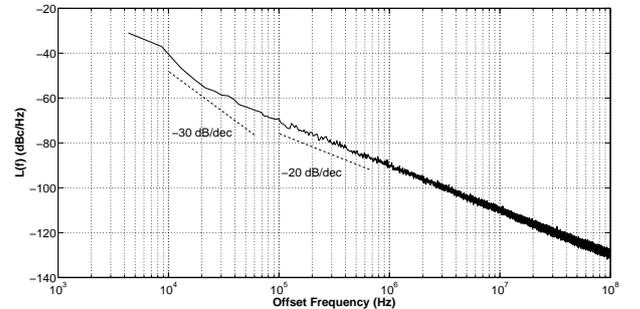


Figure 5: Open loop wreal VCO phase noise simulation result.

The phase noise specification given from measurement result is -90 dBc/Hz at a offset frequency of 1 MHz and $f_0 = 1 \text{ GHz}$. The VCO model uses `dist_normal()` to map the white noise equivalent timing jitter. Based on the theory in [10], the $1/f$ noise can be computed by passing a white noise through a filter with the transfer function:

$$H(s) = \frac{1}{s} \quad (12)$$

The transfer function can be realized in time domain based on the nonuniform sampled filter approach showed in section 4.1. Listing 2 shows the part of the accumulating jitter implementation in the wreal VCO respect the influence of flicker ($1/f$) noise.

```

module wrealVCO(out, in)
output out;
input in;
wreal out, in;
//... parameters
//... configuration check
always@(in) begin
//calculate the frequency based on input voltage
end
always begin
#next
...
dt=accjitter*$dist_normal(seed, 0,1);
dtf=dt*fc*next+dtf_old;
next=0.5/freq+dT+dTf_old;
...
end
assign out= vco_out;
...
endmodule

```

Listing 2: Part of the wreal VCO source code.

The simulation of 10^6 oscillation cycles with wreal VCO model needs about 31 seconds to complete while the simulation with the phase model used in [5] with the same specification and condition needs about 8 minutes. Figure 6 shows the comparison of the results of phase noise simulation between wreal model and phase model.

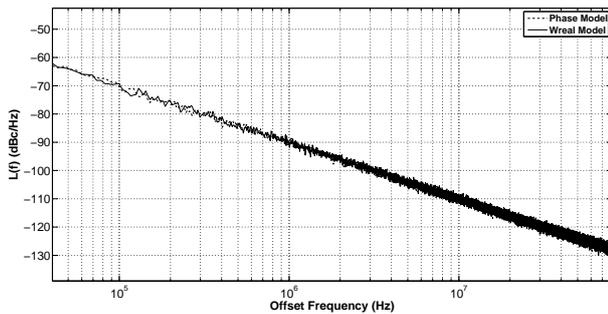


Figure 6: Phase noise comparison between VCO phase model and proposed wreal model.

Figure 6 clearly demonstrates that the wreal model can archive higher simulation efficiency while keeping sufficient accuracy for verification purpose. Using wreal event driven simulations to speed up simulation leads to another important consideration: the random delay of the timing jitter is quantized to the timing accuracy settings of the digital sim-

ulator and the implementation. This is shown in figure 7.

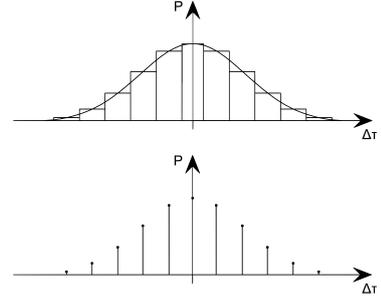


Figure 7: Quantizing effect on normal distribution.

Based on our Simulation and the calculation in [11], a specified timing jitter up to $10fs$ can be modeled accurately enough with a quantized time step of $1fs$, which is the highest time resolution setting in ncsim simulator [12].

4.3 Divider

The divider shown in figure 1 reacts on falling and rising edges of the VCO output signal. Since the wreal signal doesn't offer edge detection, the input signal has to be converted into logic value. One possibility to do this is to use an internal wire to convert the wreal signal, as shown in listing 3 shows. The counter implemented in the model increments by one when the VCO signal creates a rising or falling edge. The divider output will remain unchanged until the counter has reached the divider ratio. Additionally, the synchronous jitter is added to the output in order to map the circuit level specification.

```

module divider_wreal( out, in, dsm_in ) ;
input in ; wreal in ;
input dsm_in; wreal dsm_in;
output out; wire out;
parameter real vlo, vhi;
... parameters of jitter and divider ratio..
integer counter, fp, dSeed, dratio;
wire vin_intern ;
reg vintern ;
real diff, prev, now, dT;
assign out = vintern;
assign vin_intern=(in==vhi ? 1'b1 : 1'b0);
initial begin
...
end
always@(posedge vintern or negedge vintern) begin
if (counter == dratio-1) begin
dT=syncjitter*$dist_normal(dSeed,0,1) ;
counter = 0 ;
#((td+dT)/1f) vintern=~vintern;
end
else
counter = counter + 1 ;
end

always@(dsm_in) begin
//change the divider ratios
end
endmodule

```

Listing 3: Part of the divider source code.

The output signal of the divider serves also as clock signal for the $\Delta\Sigma$ modulator which changes the divider ratio in order to produce its fractional part.

4.4 $\Delta\Sigma$ modulator

The $\Delta\Sigma$ modulator used in the fractional-N PLL minimizes the phase noise close-by the center frequency by pushing the phase error towards higher frequency bands. Due to its pure digital implementation on circuit level, the model of $\Delta\Sigma$ modulator can be realized using event driven approach based on its transfer function showed in figure 8. The 2 MSBs of the quantizer output are used to set the divider ratio. Listing 4 shows the source code of the wreal $\Delta\Sigma$

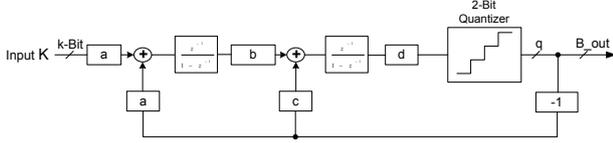


Figure 8: 2nd order $\Delta\Sigma$ modulator used in the PLL.

modulator, where the dithering signal is applied into the first register of the modulator to improve the noise performance.

```

module DSM (out, out_quant, clk, K, dithin);
output out, out_quant;
input clk, K;
wreal dithin, k, out, out_quant;
parameter real a;
parameter real b;
parameter real c;
parameter real d;
real reg1, reg2, temp_out, max_quant, temp_out_quant;
initial begin
max_quant=pow(2, 16.0);
reg1=0;
reg2=0;
temp_out=0;
temp_out_quant=0;
end

always@(posedge clk) begin
temp_out=reg2*d;
temp_out_quant=
temp_out-(temp_out-(max_quant*
(floor(temp_out/max_quant))));
reg2=(reg1*b)+((-temp_out_quant)*c)+reg2;
reg1=reg1+((-temp_out_quant)*a)+(K)+dithin;
end // end of always @(posedge clk)

assign out=temp_out_quant;
endmodule

```

Listing 4: Example source code for the 2nd order $\Delta\Sigma$ modulator.

The noise contribution of the $\Delta\Sigma$ modulator to the over all PLL phase noise is shown in figure 9. The spurious are in multiples of the reference frequency, which is 13 MHz.

4.5 PFD And Charge Pump

Based on circuit level simulation and the theory from [13], the Phase-Frequency-Detector (PFD) can not detect small phase errors between reference and divider output signals. Thus it loses its linearity in small phase errors. The $\Delta\Sigma$ modulator is very sensitive to such non-linearity and responds with increased phase noise and spurious emissions. Additional delay elements shown in figure 10 has been inserted between the “AND” gate and the reset signal input of

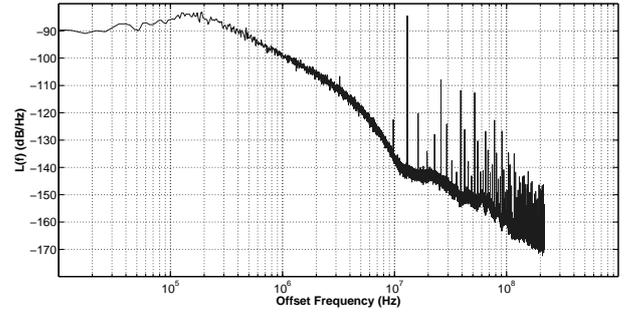


Figure 9: Phase noise contribution of the 2nd order $\Delta\Sigma$ modulator used in the PLL.

the flip flops in order to reduce the influence of small phase errors.

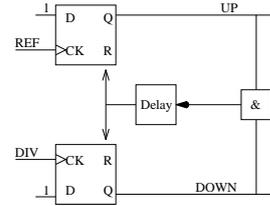


Figure 10: The PFD structure.

This also leads to a reduced linear range of the PFD [13]. So only phase errors between 0 and $2\pi - \Delta$, where $\Delta = \frac{2\pi \cdot \tau}{T_{ref}}$, can be corrected. The mismatch of the up/down current from the Charge Pump (CP) is responsible for reference spurious at the output. For the verification of the overall PLL phase noise performance, the aforementioned non-ideal behavioral from PFD and CP respectively their synchronous jitter have to be mapped properly. With the new modeling and simulation approach, the PLL system shown in figure 1 is going to be simulated and verified in the next section.

5. DEMONSTRATION EXAMPLE

The following specifications are derived from a typical mixed signal fractional-N PLL frequency synthesizer, partly based on circuit level simulations on single, isolated blocks. The

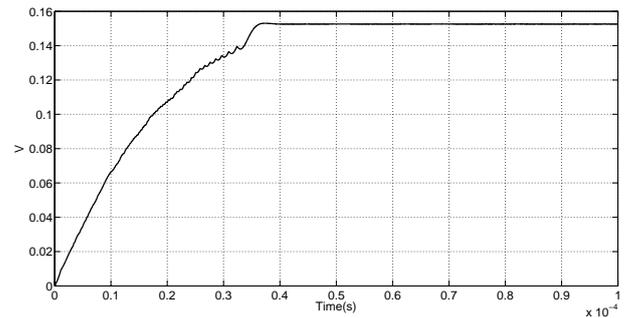


Figure 11: PLL lock-in process.

reference frequency is 13 MHz. The output signal frequency of the VCO is about 433 MHz with a phase noise of -90 dBc/Hz at 1 MHz offset. The tuning range of the VCO is between 368 MHz to 450 MHz. The phase noise of

the charge pump is about 80.77 dBc/Hz at 25 kHz offset. Figure 11 shows the lock-in process of the PLL. Its duration is about 40 us which is close to the measurement on real silicon. The overall phase noise measurement is only available for three offset frequency points shown in figure 12 and table 1.

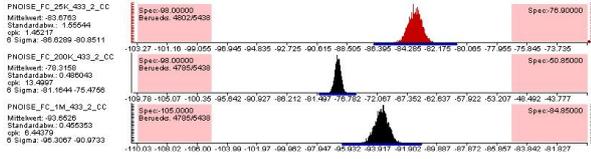


Figure 12: Chip measurement results.

Offset frequency	25 KHz	200 KHz	1 MHz
Simulation result (dBc/Hz)	-84	-75	-91.5
Measurement result (dBc/Hz)	-83.7	-78	-93
Δ (dB)	0.3	3	1.5

Table 1: Comparison between model simulation and measurement on real silicon results.

The maximum difference between our model level simulations and the measurements is about 3 dB, which makes the accuracy of the model absolute sufficient enough for the functional verification with a rough performance estimation.

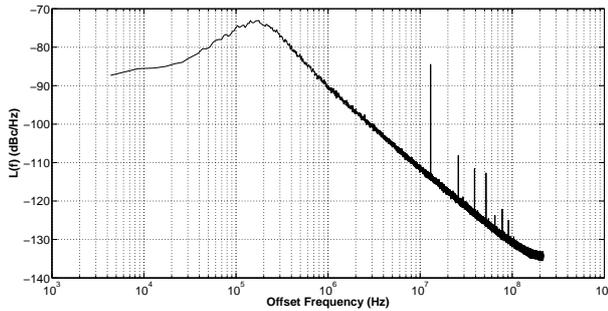


Figure 13: The overall phase noise of the PLL system based on model simulation.

While the top level simulation on circuit level could take more than a week to complete, without the consideration of convergence and memory problems, the verification based on the wreal models took less than one hour to complete the phase noise simulation and configuration check.

6. CONCLUSION

In this paper, we have presented the event driven analog modeling approach for the fractional N frequency synthesizers. The models provide a noticeable speedup even against the widely used phase domain models. The explicit mapping of the single block specification leads to a parameterizable top level PLL test bench. This provides the possibility to verify and optimize routines and distributions of specifications, based upon an identical test bench throughout the whole design and verification process. During the simulation, the complete mapping of phase noise, nonlinearity and digital controlling have been taken into account. The implementation of the proposed approach into an available design flow can lead to a fast and reliable functional verification process.

Acknowledgement

The authors would like to thank the Cadence Academic Network (CAN) for the software support used for this work. Results from a research project of the German Government called VeronA (BMBF 01 M 3079) have influenced this work.

7. REFERENCES

- [1] H. Chang and K. Kundert. Verification of Complex Analog and RF IC Designs. *Proceedings of the IEEE*, 95:622–639, 2007.
- [2] K. Kundert. Challenges in RF simulation. In *Radio Frequency integrated Circuits (RFIC) Symposium, 2005. Digest of Papers. 2005 IEEE*, pages 105–108, 2005.
- [3] S. Joeres, H.-W. Groh, and S. Heinen. Event driven analog modeling of RF frontends. In *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International*, 2007.
- [4] Shuilong Huang, Huainan Ma, and Zhihua Wang. Modeling and simulation to the design of Fractional-N frequency synthesizer. In *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE '07*, pages 1–6, 2007.
- [5] Zhipeng Ye, Wenbin Chen, and Michael Peter Kennedy. Modeling and simulation of DeltaSigma Fractional-N PLL frequency synthesizer in Verilog-AMS. *IEICE Trans Fundamentals*, E90-A(10):2141–2147, Oct 2007.
- [6] Ken Kundert. Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers. *Designers Guide Community*, 2006.
- [7] R.B. Staszewski, C. Fernando, and P.T. Balsara. Event-driven simulation and modeling of phase noise of an RF oscillator. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(4):723–733, 2005.
- [8] R.A. Cottrell. Event-driven behavioural simulation of analogue transfer functions. In *Design Automation Conference, 1990. EDAC. Proceedings of the European*, pages 240–243, 1990.
- [9] A.A. Abidi. Phase Noise and Jitter in CMOS Ring Oscillators. *Solid-State Circuits, IEEE Journal of*, 41:1803–1816, 2006.
- [10] Jongsun Park, K. Muhammad, and K. Roy. Efficient modeling of $1/f^\alpha$ noise using multirate process. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(7):1247–1256, July 2006.
- [11] Y. Wang, S. Joeres, R. Wunderlich, and S. Heinen. Modeling approaches for functional verification of RF-SoCs: limits and future requirements. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(5):769–773, 2009.
- [12] Cadence. Cadence verilog-ams language reference. volume 8.1, May 2008.
- [13] Kun-Seok Lee, Byeong-Ha Park, Han il Lee, and Min Jong Yoh. Phase frequency detectors for fast frequency acquisition in zero-dead-zone cpplls for mobile communication systems. pages 525–528, Sept. 2003.