A Fast and Accurate SystemC-AMS Model for PLL

Kezheng Ma, Rene van Leuken
Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Delft, The Netherlands
K.Ma@students.tudelft.nl

Maja Vidojkovic, Jac Romme, Simonetta Rampu, Hans Pflug, Li Huang, Guido Dolmans
Holst Centre/imec
Eindhoven, The Netherlands

Abstract—**PLLs have become an important part of electrical systems. When designing a PLL, an efficient and reliable simulation platform for system evaluation is needed. However, the closed loop simulation of a PLL is time consuming. To address this problem, in this paper, a new PLL model containing both digital and analog parts based on the recently published SystemC/SystemC-AMS (BETA version) language is presented. Many imperfections such as Voltage Control Oscillator (VCO) noise or reference jitter are included in this model. By comparing with the Matlab model, the SystemC/SystemC-AMS model can dramatically reduce simulation time. Also, by comparing with Analog Device's ADI SimPLL simulation results, Cadence simulation results and real measurement results, the accuracy of the SystemC/SystemC-AMS model is demonstrated. The paper shows the feasibility of a unified design environment for mixed-signal modeling based on SystemC/SystemC-AMS in order to reduce the cost and design time of electrical systems.**

Index Terms—**SystemC/SystemC-AMS; Phase Locked Loop; mixed-signal modeling; hardware description language.**

I. Introduction

Phase Locked Loops (PLL) have become an important part of electrical systems. It is widely used in radios, computers, telecommunications, and other electronic applications. However, when designing a PLL, one of the main problems is that the closed loop simulation of a PLL is very time consuming, especially at transistor level. For example, a fine frequency resolution required in noise simulation often leads to an over 24-hour simulation time [1]. Therefore, to examine the closed loop behavior, a fast and precise model is highly desirable.

Many efforts have been spent to address this problem, such as in [1] and [2] where Verilog-AMS models are presented. However, these models have their limitations in terms of simulation performance. For instance, in [1], the high frequency output of the Voltage Control Oscillator (VCO) is not generated in order to reduce simulation time. But in this way, it is impossible to insert the PLL into a radio system model for further testing. In [2], although the VCO output is generated, performing a 60us transient analysis requires 15 minutes. Consequently, the simulation time will be over 4 hours if a transient analysis over 1000us (for noise analysis) is needed.

To further improve simulation time while still keeping good accuracy, a new mixed-signal PLL model based on the recently published language SystemC/SystemC-AMS (BETA version) is proposed in this paper. Unlike many other high center frequency models, the model in this paper does not combine the divider and the VCO. Consequently, the high frequency VCO output is available, which enables designers to insert the PLL into a radio system model to evaluate the overall system performance. In the past, we published one paper on SystemC/SystemC-AMS PLL model [3]. However, this model only has basic functionality and includes limited practical imperfections. To develop a more realistic model for system design, many imperfections are included in the model proposed in this paper, e.g. VCO noise, PLL Dead Zone, Phase Frequency Detector (PFD) reset delay, Charge Pump (CP) current mismatch, CP leakage current, the reference jitter and Delay difference between the CP control signal. The simulation results of the proposed model are compared with the simulation (measurement) results of a Matlab model, Analog Device's ADI SimPLL tool, Cadence, and real chip implementation. The comparison results verify that the SystemC/SystemC-AMS model is precise with fast simulation time.

The paper is organized as follows. The SystemC/SystemC-AMS language is briefly introduced in section II. The structure of the model is discussed in section III. The behavioral modeling of each PLL block is shown in section IV. Our proposed model is verified in section V. Finally, conclusions are given in section VI.

II. A brief introduction to SystemC / SystemC-AMS

SystemC/SystemC-AMS is an effective system-level simulation and modeling language, which offers many advantages. First of all, it is a C++ based language which ensures fast calculation speed. It is also very easy to take advantage of C++ power by using a large choice of existing C++ libraries. Secondly, in Timed Data Flow model in SystemC-AMS [4], it is allowed to set different time steps to different blocks which enable designers to set high resolution to some important blocks while lower resolution to other blocks. Last but not least, advanced tracing mechanisms are available to enable or disable time-domain or frequency-domain tracing while running simulations.

A. SystemC

SystemC does not have the ability to model analog circuits. It is used to model the digital circuits. In our proposed model here, SystemC is used to model digital blocks.

B. SystemC-AMS

In order to simulate the analog circuits, AMS extensions are introduced based on SystemC. The SystemC-AMS extensions are built on top of the SystemC language standard IEEE 1666-2005 [4]. SystemC-AMS can be further divided into 3 models -- Timed Data Flow model (TDF), Linear Signal Flow model (LSF), Electrical Linear Networks model (ELN). For detailed information about these models, please refer to [4].

C. Simulation flow

The simulation flow is shown in Figure 1. The digital parts of the system are modeled with SystemC while the analog parts are modeled with SystemC-AMS. After transient simulations, the generated data will be loaded into Matlab to perform frequency analysis.

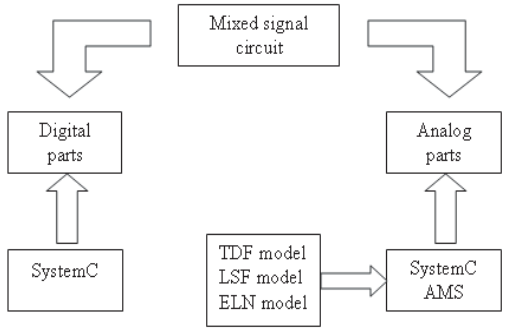

Figure 1 SystemC/SystemC-AMS model design flow.

III. Model Structure

In radio transmitters, an integer N-PLL is used to synthesize new frequencies which are multiples of a reference frequency, with the same stability as the reference frequency. The structure of the integer PLL modeled in this paper is shown in Figure 2, in which the red and blue parts are modeled with SystemC and SystemC-AMS, respectively.

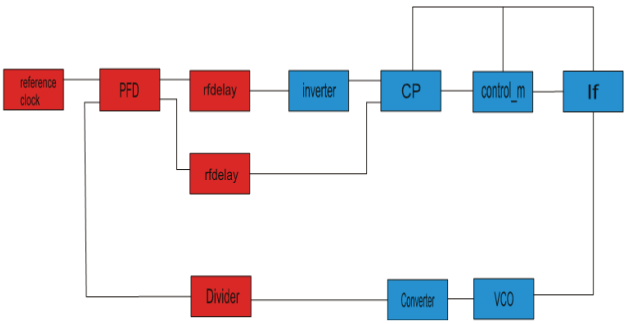

Figure 2 Structure of the PLL model.

As shown in Figure 2, the model is composed of eleven blocks, a reference clock block, a phase/frequency detector block, two rfdelay blocks, an inverter block, a charge pump block, a control_m block, a loop filter (LF) block, a voltage control oscillator block, a converter block and a divider block. The reference clock block is used to generate reference signal, the phase/frequency detector (PFD) block is used to detect the phase/frequency difference between the reference signal and the divider output. After the PFD, the signal goes into rfdelay block and inverter to generate the signals that control the charge pump. The CP will charge the loop filter to generate the VCO control voltage. The control_m block is used to limit the output voltage of the loop filter. The converter block transfers the VCO analog output signal to a digital input signal that suitable for the frequency divider.

IV. Circuit Model Based on SystemC/SystemC-AMS

In this section, the behavior SystemC/SystemC-AMS model of each block will be presented.

A. Reference clock block

The reference block is used to generate the reference clock of the PLL. Part of the code of reference block is shown in Figure 3. It works as follows: assume the period of the reference clock is T, then every T/2, the output will reverse its value. The function "gauss()" is used to generate a gauss random number with 0 mean and variance of 1.0e-9 which is used to model the reference jitter.

```
while(1)
{
        a=!a;
        wait(ref+gauss(1.0e-9),SC_US);
        a=!a;
        wait(ref+gauss(1.0e-9),SC_US);
}
```

Figure 3 Part of source codes of the reference clock block.

B. The PFD block

The structure of the PFD is shown in Figure 4, it is composed of 2 D-registers, a delay block and an and-gate. The PFD inputs signals, reference clock and divider output, serve as the clock of the registers. The D ports of the registers are connected to the "VDD" supply. The registers' outputs qa and qb are the outputs of PFD. If qa and qb are both 0 and there is a rising edge in reference clock, qa will become 1. If this follows by a rising event in divider output, qb will become 1 and the registers are reset. A delay block is introduced to model the reset-delay. Also it can be used to eliminate the dead zone. As shown in Figure 5, it is clear that the phase and frequency difference is detected. If the reference frequency is larger than that of divider output or reference phase leads to the divider output phase, there is a pulse in qa and qb is equal to 0 (maybe with very small pulse because of the reset delay). On the contrary, if the reference frequency is smaller than that of divider output or reference phase delayed from the divider output phase, there is a pulse on qb and qa is equal to 0.

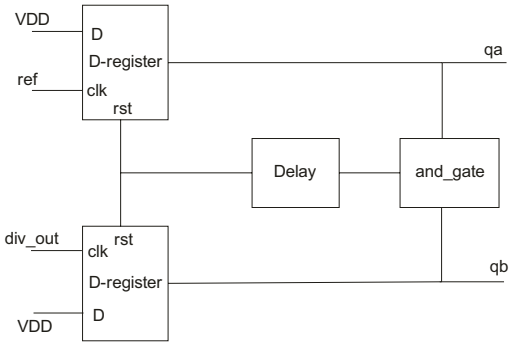

Figure 4 Structure of the PFD.

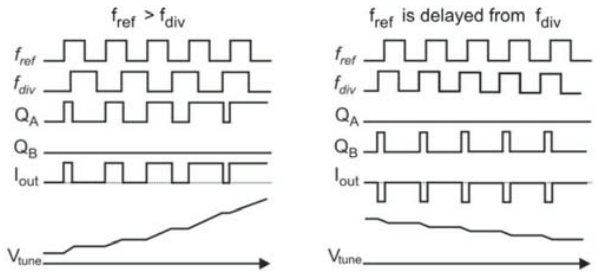

Figure 5 PFD waveform [5].

C. The rfdelay block

Both the rising edge and the falling edge of a digital signal are non ideal, i.e. there is some delay when the signal changes from "1" to "0" or from "0" to "1". If the pulse width of the PFD outputs is too small, the signals' value will not reach its peak value. This causes PLL Dead Zone. The rfdelay block is used to model this behavior. Figure 6 shows part of the code of this block. It acts as follows: when the input is 1 or 0, the output will increase or decrease gradually (0.03 V per 0.1 ns) until it is larger than 1.199 V or smaller than 0.001 V.

```
        wait(clear_in.value_changed_event());
        if(clear_in==1)
        {
b:      while(clear_out<1.199){
                clear_out=clear_out+0.03;
                wait(0.1,SC_NS);
                if(clear_in==0){goto a;}
        }
        }
        else
        {
a:      while(clear_out>0.001){
                clear_out=clear_out-0.03;
                wait(0.1,SC_NS);
                if(clear_in==1){goto b;}
        }
```

Figure 6 Part of source codes of the rfdelay block.

D. Inverter

It should be noticed that the inverter includes the rising and falling delay. The delay will contribute to the delay difference between the Charge Pump control signals. The delay difference causes ripples in the VCO control voltage and further causes spurs (reference spurs) in the VCO output spectrum by frequency modulation.

E. Charge pump and loop filter

The structure of the charge pump and loop filter are shown in Figure 7. The outputs of the PFD (after the rfdelay and inverter) are connected to gate Mp1 and Mp2. If the frequency of reference clock is higher, the Mp1 will open and the up current source will charge the loop filter. As a result, the tuning voltage "Vtune" will increase and the VCO output frequency will increase. Similarly, if the reference frequency is lower, the VCO output frequency will decrease. Figure 8 shows part of the code of the charge pump. The current sources of the charge pump are transistors working in saturation region. When the voltage Vtune is larger than 0.3V and smaller than 0.9V, both current sources work in saturation region, the charge current and discharge current will be equal to the saturation current "currentup", "currentdw" respectively. The variable "mis" indicates the current when both switches are closed which is the charge pump current mismatch. When Vtune is smaller than 0.3V, the down current source goes into the triode region and the current provided by it will be smaller. In the same way, when Vtune is larger than 0.9V, the up pump goes into the triode region and the up pump current will become smaller. The other imperfection modeled in this charge pump model is the leakage current, when both switches are open (a>0.6&&b<0.6), there is still a small negative current exist (leakage).

In summary, there are 3 non-ideal effects modeled in the Charge Pump model, current mismatch, transistor triode region and leakage current. The current mismatch and leakage current cause the reference spurs. Entering transistor triode region increases the settling time.

The loop filter is a second order loop filter with two capacitors and one resistor and is modeled with the ELN model.

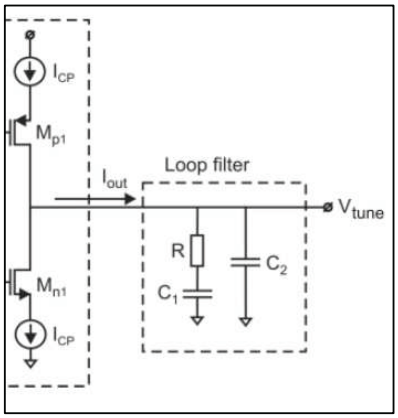

Figure 7 Structure of the charge pump and loop filter [5].

```
if(vtune>0.3&&vtune<0.9)
{
        charge=currentup;
        discharge=-currentdw;
        oneone=charge+discharge;
}
else if(vtune<0.3)
{
        vds=vtune;
        charge=currentup;
        discharge=-
20/9.0*currentdw*vds+100/81.0*currentdw*vds*vds;
        oneone=charge+discharge;
}
else if(vtune>0.9)
{
        vds=1.2-vtune;
        charge=20/9.0*currentup*vds-
100/81.0*currentup*vds*vds;
        discharge=-currentdw;
        oneone=charge+discharge;
}

if(a<0.6&&b<0.6)
{out=charge;}
else if(a>0.6&&b>0.6)
{out=discharge;}
else if(a>0.6&&b<0.6)
{out=-leakage;}
else if(a<0.6&&b>0.6)
{mis=oneone;}
```

Figure 8 Part of source codes of the charge pump block.

F. Control_m block

In SystemC/SystemC-AMS, the voltage has no limit. In other words, the voltage can go to infinite. The control_m block is used to limit the voltage. When the "Vtune" is larger than the supply voltage (1.2V here) or smaller than 0, control_m will cut off the current from charge pump which prevents "Vtune" keep on changing.

G. VCO

The output of VCO is given by

$$\cos(\omega 0 \times t + 2 \times \pi \times Kvco \times \int vtunedt) \qquad (1)$$

where the w_0 is free running VCO frequency, K_{vco} is the VCO gain (160MHz/V here), vtune is the control voltage of the VCO (Vtune). VCO noise is added to the output of the VCO, the way to model VCO noise is shown in Figure 9. First, a gauss random number generator creates a sequence of random number with 0 mean and variance 1. Then it is multiplied with a gain "$\sqrt{Cvco}$". After the multiplication, an integrator integrates the result to change it to the phase domain. Finally, the noise phase is added to the output phase of the VCO. Part of the VCO code is shown in Figure 10. In the code, "0.01e-9" is the time step. "in" is the VCO control voltage. "integrationrn" represented the noise phase.

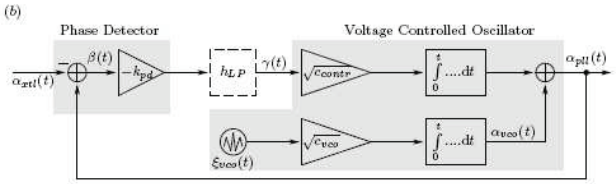

Figure 9 VCO noise model[6].

```
integrationr=integrationr+0.01e-9*gain*2.*M_PI*in;
integrationrn=integrationrn+sqrt(0.01e-9)*7*2.*M_PI *noise;
out.write( cos(
sc_time_stamp().to_seconds()*(control*2.*M_PI)+integrationr+i
ntegrationrn)) ;
```

Figure 10 Part of source codes of the VCO block.

H. Converterb

The PLL model is a mixed signal model. A converter block is used to transfer the analog signal to a digital signal. In the feedback loop, it converts the VCO sine wave output to a square wave which is suitable for the digital divider. It acts as follows, when the sine wave's value is larger than 0, the output is equal to 1, if it is smaller than 0, the output is equal to 0.

I. Divider

The divider is modeled as a counter. If there is an event in the input, a signal "inter" will add one. If "inter" is larger than a certain number N(4798 here), the output will be reversed. Part of the divider code is shown in Figure 11.

```
while(1){
        wait(in.value_changed_event());
        inter = inter.read() + 1;
        if(inter.read() > 4798){
                out=!out.read();
                inter.write(0);
        }
```

Figure 11 Part of source codes of the divider block.

V. Simulation Results

The model proposed in this paper has been used to simulate the integer PLL represented in [5]. The PLL has a reference clock of 500kHz, an output frequency of 2.4GHz, a charge pump current of 50uA. The leakage current is set to 25pA and the charge pump current mismatch is 5uA. The simulation time for 1200us transient analysis with a 0.01ns resolution is about 15 minutes.

To ensure the correctness of the model, the simulation results are compared with the Matlab simulation results, Analog Device's ADI SimPLL simulation results, Cadence

simulation results, and the measurement results. The parameter values in ADI SimPLL, Cadence and real circuits are equal to the values mentioned above. When comparing with the Matlab model, the parameters are set as follows: a reference clock of 31.2MHz, an output frequency of 124.8MHz, an up pump current of 25uA and a down pump current of 15uA. The free running VCO frequency is 124MHz.

A. Comparison with Matlab simulation results

The simulation result of the SystemC/SystemC-AMS model is first compared with the simulation results of a Matlab model. The Matlab model is a sample-driven simulator for charge-pump PLLs. To shorten simulation-time, the VCO output signal is modeled in the phase domain, such that the frequency division becomes a mere scaling of the phase. The phase noise is modeled as a Gaussian random walk process and the current-source imbalance can be simulated.

The two models have the same parameter values and include the same imperfections. Figure 12 shows the VCO output power spectrum density (PSD) of the two models, in which the red line is the PSD of the SystemC/SystemC-AMS model while the blue one is that of the Matlab model. As shown, the two results are very similar.

Figure 13 and Figure 14 show the time domain simulation results of VCO control voltage. The settling time of the Matlab model is about 1.5 us whereas that of the SystemC/SystemC-AMS model is about 1.5 us too. It is clear that our model and the Matlab have comparable simulation results. The simulation time of SystemC/SystemC-AMS model (about 2 minutes for 650 us simulation with a resolution of 0.1 ns) is about 10 times smaller than that of the Matlab model (about 20 minutes for 639 us simulation with a resolution of 0.1 ns).

Another advantage of SystemC/SystemC-AMS over Matlab is its capability to simulate very high frequency signal in a high resolution. For example, for a 2.4GHz signal with resolution of 0.01ns, SystemC/SystemC-AMS is able to simulate it while Matlab fails (Matlab will run out of memory).

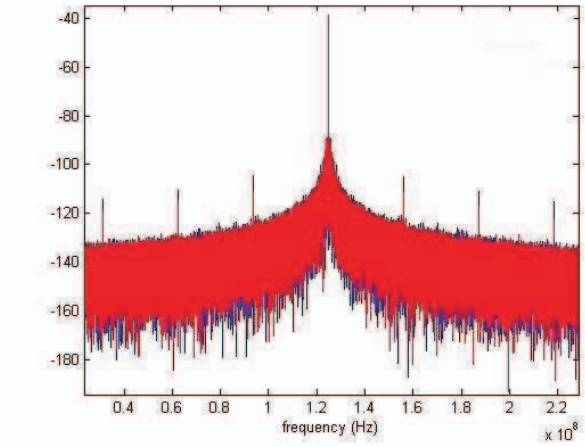

Figure 12 PSD of VCO output of our and Matlab models.

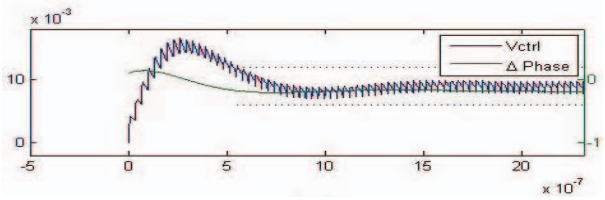

Figure 13 VCO control voltage of the Matlab model.

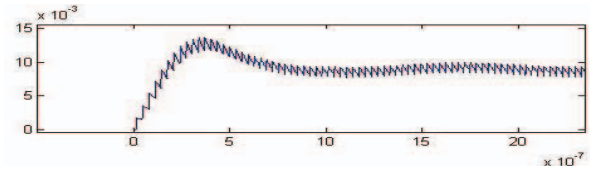

Figure 14 VCO control voltage of our model.

B. Comparison with ADI SimPLL simulation results

The simulation tool ADI SimPLL published by Analog Devices is used to verify the close loop simulation results of the SystemC/SystemC-AMS model. Figure 15 shows the VCO output frequency of SystemC/SystemC-AMS model and Figure 16 shows the frequency of the ADI tool's simulation result. The parameter values of both are the same. The settling time of both results is about 60 us and the waveforms are comparable. Thus, the closed loop behavior is correctly modeled with SystemC/SystemC-AMS.

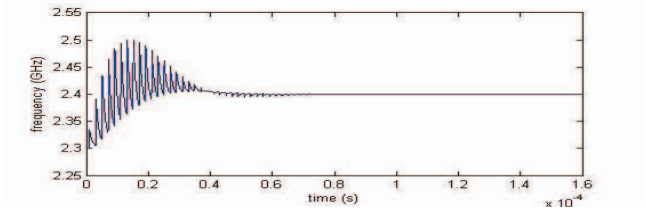

Figure 15 VCO output frequency of our model.

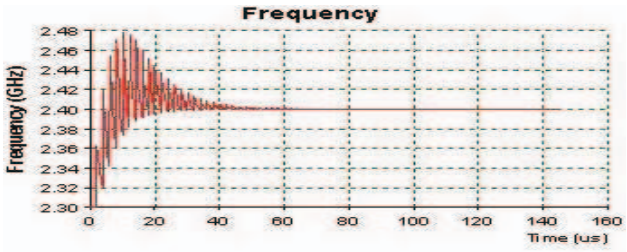

Figure 16 VCO output frequency of the ADI SimPLL tool

C. Comparison with Cadence simulation results

The simulation results of the PLL closed loop behavior were verified in the previous subsection. In this section, the open loop simulation results are verified by Cadence simulation results. Figure 17 and Figure 18 show the open loop simulation results of Cadence and the SystemC/SystemC-AMS model respectively. The input signals of the two models

are same. In the figure, the first two signals are the outputs of PFD, the third signal is VCO control voltage Vtune, the last signal is the charge pump current. As can be seen from the figure, in the 18th cycle, the value of Vtune of Cadence simulation results is 403.9 mV and that of SystemC/SystemC-AMS model is 417.0 mV, which are similar. The other signals are also similar, except that in the Cadence results, there are some current spurs when the current source goes into the triode region. Normally the current sources will not enter the triode region. Thus these spurs are not modeled here in order to further decrease simulation time.

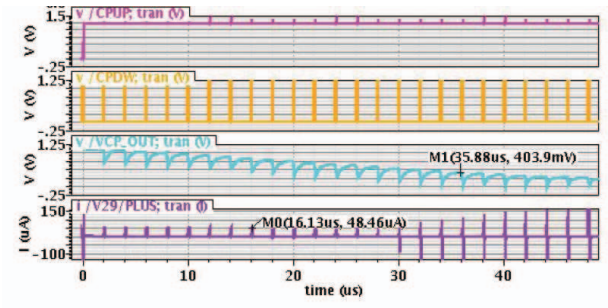

Figure 17 Open loop simulation results of the Cadence.

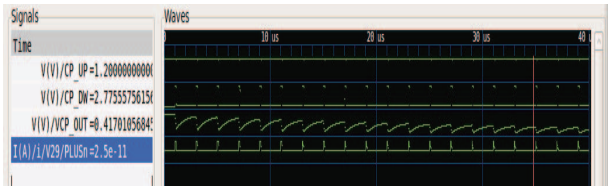

Figure 18 Open loop simulation results of our model.

D. Comparison with measurement results

Figure 19 shows the measurement PSD of the Pre-Scaler output (VCO output signals divided by 30) of the PLL in [5] while Figure 20 shows that of the SystemC/SystemC-AMS model. From these two figures, it is clear that the model can correctly reflect the real PLL noise behavior.

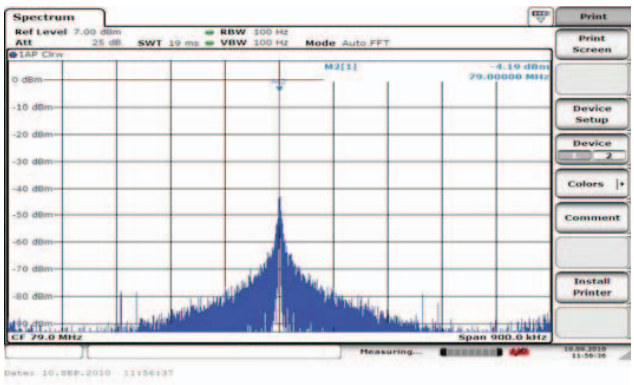

Figure 19 Measurement result of the prescaler output PSD [4].

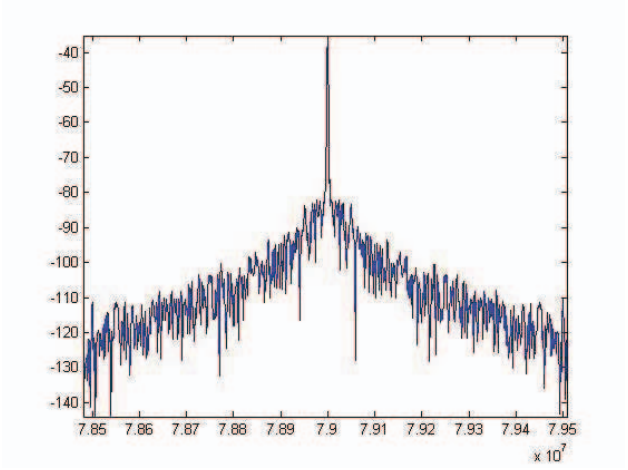

Figure 20 Simulation result of the prescaler output PSD of our model.

VI. Conclusion

A new and precise mixed-signal model based on the recent published language SystemC/SystemC-AMS is presented in this paper. The simulation time is significantly less than that of the Matlab model. The SystemC/SystemC-AMS model gives the insights of the key characteristics determining overall system performance. Moreover, the effects of various practical imperfections are included in this model. The SystemC/SystemC-AMS model is verified by comparing the simulation results with a Matlab model, ADI SimPLL simulation tool, the Cadence simulations and a real chip implementation. These comparison results demonstrate that the model is able to precisely reflect the real system behavior with fast simulation time. They also validate that the SystemC/SystemC-AMS is an efficient and powerful language for mixed-signal modeling.

Acknowledgment

The authors would like to acknowledge the support from the Beyond Dreams Project.

References

[1] S. Huang, H. Ma, and Z. Wang, “Modeling and Simulation to the Design of ΣΔ Fractional-N Frequency Synthesizer”, in Design, Automation & Test in Europe Conference & Exhibition, Nice, Apr. 2007, pp. 1-6.

[2] L. Liu, Y. Yang, Z. Zhu, and Y. Li, “Design OF PLL System Based VERILOG-AMS Behavior Models” in VLSI Design & Video Tech, May 2005, pp. 67-70.

[3] T. Xu, H.L. Arriens, R. van Leuken, amd A. de Graaf, “A precise SystemC-AMS model for Charge Pump Phase Lock Loop with multiphase outputs”, in ASICON, Oct. 2009, pp. 1-6.

[4] Open SystemC Initiative (10-02-2010), “SystemC AMS extensions User’s Guide” [online]. Available: http://www.systemc.org/ downloads/ standards/

[5] P. Harpe, C. Huang, S. Rampu, and M. Vidojkovic, “Analog BAN Radio Blocks – Designs for Oct09 and April10 Tapeouts”, Holst Centre, Netherlands, Tech. Rep. TN-10-WATS-TP2-050, Jun. 2010.

[6] S. Bittner, S. Krone, and G. Fettweis, “Tutorial on Discrete Time Phase Noise Modeling for Phase Locked Loops” [online]. Available: http://www.vodafone-chair.com/staff/bittner/.