Periodic Steady State Simulation of Mixed-Signal RF Circuits

Dušan N. Grujić, Mihajlo Božović, Pavle Jovanović, Milan Savić, and Lazar Saranovac

Abstract – Advances in CMOS technology have enabled the use of digitally assisted RF concepts for significant performance improvement and cost reduction. As a result, periodic steady state simulation of mixed-signal RF circuits is becoming increasingly important. Standard mixed-signal simulators only support transient analysis, while RF simulators are not supporting digital simulation and have problems with memory elements. In this paper we propose a solution for periodic steady state simulation of mixed-signal RF circuits. Advantages, limitations and pitfalls of the proposed solution are presented. The proposed solution is demonstrated on example of dynamic performance improved RF D/A converter simulation. Significant improvement of SFDR shows the importance and potential of digitally assisted RF circuits, and the ability to efficiently simulate them.

Keywords - periodic steady state, mixed-signal, RF

I. Introduction

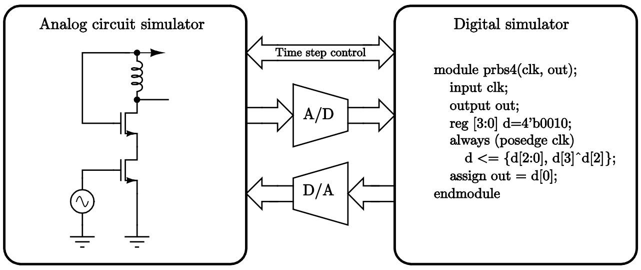
Circuit simulation requires a delicate balance between accuracy, simulation time and resources. Higher level abstraction methodologies and tools have been developed to cope with ever growing complexity, while preserving the required accuracy and important metrics. These concepts have been adopted in digital design decades ago, where an enormous number of gates made a transistor level simulation impractical, and in many cases impossible. Higher abstraction levels have been widely adopted in analog design much later, with the advent of Verilog-A [1]. Further increase in simulation efficiency has been achieved by the development of specialized types of simulation, such as periodic steady state (PSS) for RF circuits, and simulators, such as mixed-signal simulators.

Increased use of digital gates in RF circuits, e.g. [2], has created the need for PSS simulation of mixed-signal circuits. Currently available mixed-signal simulators are not capable of PSS simulation, while the PSS analysis does not support simulation of digital circuits "out of the box", and suffers from hidden state problem [3].

This paper reviews the limitations of mixed-signal and PSS simulations in Sec. II, and proposes a solution for the functional mixed-signal PSS simulation in Sec. III. Demonstration of proposed solution is presented in Sec. IV. Conclusion and final remarks are given in Sec. V.

II. Mixed-Signal and Periodic Steady State Simulation Limitations

Mixed-signal simulation is performed by coupling analog and digital simulators with A/D and D/A converters, and simulating with the common time step control, as shown in Fig. 1. At each time step analog simulator uses the Newton-Raphson method to solve the non-linear circuit, while the digital simulator performs an event-driven simulation of the compiled Verilog/VHDL/SystemC or mixed-language design. This way both analog and digital simulators can generate events and schedule new time steps for simulation.



```
module prbs4(clk, out);
   input clk;
   output out;
   reg [3:0] d=4'b0010;
   always (posedge clk)
      d <= {d[2:0], d[3]^d[2]};
   assign out = d[0];
endmodule
```

Fig. 1. Analog and digital simulator coupling in mixed-signal simulation

Analog simulator time step is determined by local truncation error (LTE) algorithm and scheduled breakpoints, e.g. by independent sources. Digital simulator can schedule a new time step in two ways: generating an event by behavioral code, and by simulating the propagation delay. The important difference is that behavioral code can generate events on its own, without external stimuli, while the propagation delay generates new events only on external stimuli. If the digital design is synthesizable, it cannot generate events without external stimuli. Furthermore, if a functional model is simulated there are no propagation delays, and the digital simulator cannot schedule new time points at all. Therefore, time step of a mixed-signal simulation containing a synthesizable functional digital design is determined solely by the analog simulator. Digital simulator outputs are only evaluated at time points determined by the analog simulator.

Mixed-signal simulation is efficient, but has its drawbacks, both technical and practical. From the practical point of view, correct setup of mixed signal simulation is all but trivial. There is a cost issue as well, since licenses for both analog and digital simulators are needed.

D. Grujić, M. Božović, P, Jovanović and L. Saranovac are with School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11000 Belgrade. D. Grujić, M. Božović, P. Jovanović and M. Savić are with Lime Microsystems DOO, Bul. Mihajla Pupina 10a, 11070 N. Belgrade. E-mail: {d.grujic, m.bozovic, p.jovanovic, m.savic}@limemicro.com.

From a technical point of view, mixed-signal simulation is severely limited, since it only supports transient, AC and linear noise simulations. Transient simulation impedes the analysis of PSS response of RF circuits in several ways:

- Long simulation time might be needed for stiff circuit to enter steady state [4],
- Detection of steady state can be difficult,
- Even when the steady state is reached and detected, the problem of calculating the spectrum remains. Non-uniform time step is not suitable for FFT, but using a fixed time step might skip important transitions, introduce artificial jitter, violate the tolerance settings, or prohibitively prolong the simulation time,
- Large signal steady state transfer functions and noise folding cannot be calculated.

These drawbacks were exactly the motivation for the development of specialized PSS analysis. The usefulness of transient-only mixed-signal simulation of RF circuits is limited to functional verfication.

PSS simulation is suitable for analysis of RF circuits, but does not support the simulation of digital circuits "out of the box". The problem of digital circuits in PSS simulation is not simulator specific, but is related to the mathematical formulation of shooting methods used to determine the steady state, and the inherent hidden state problem [3].

Hidden state problem can be alleviated by exposing the memory element variable to the analog solver by making it an electrical quantity, e.g. charge on a capacitor. Such approach has been taken in [3] for making Verilog-A models of D flip-flop and frequency counter suitable for PSS simulation. However, this approach has two drawbacks:

- Method is not general, and requires case by case consideration,
- Even if all of the memory element models are available in Verilog-A, each instance increases the size of circuit matrix. Having in mind that digital circuits can easily have thousands of memory elements, simulation time and memory requirements might be unacceptable.

In principle, limitations of mixed-signal and PSS could be solved by simulating the complete design on a transistor level. However, this would result in prohibitively long simulation time and memory usage, and possibly convergence difficulties, due to excessively large number of transistors. A solution which allows the use of PSS analysis and has the efficiency of mixed-signal simulation is proposed in the next Section.

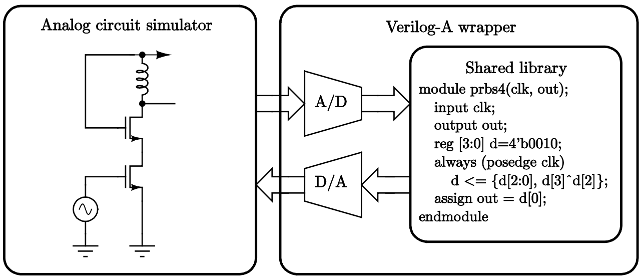
III. Proposed Solution

In a mixed-signal simulation, synthesizable digital circuit without propagation delays (functional model) does not generate new time points, and is evaluated only at time points determined by the analog simulator. This fact eliminates the need for event scheduling and notion of time in the digital simulator, greatly simplifying its design. Such a simulator can be generated from Verilog code by using an open source tool Verilator [5][6][7].

Verilator generates a digital simulator in C++ of a given module written in synthesizable subset of Verilog language. It has been successfully used in industry for generating cycle-accurate microcontroller models [8]. Simulator generated by Verilator does not have an event scheduler, and the outputs are evaluated only when requested, making it a perfect fit for a functional mixed signal simulation of synthesizable Verilog code. Cadence Spectre circuit simulator allows the use of user-defined C functions in Verilog-A modules. This feature can be used as an interface to digital simulator generated by Verilator.

Usability of synthesizable digital circuit without propagation delays requires some justification. The requirement that a digital circuit should be synthesizable is not a restriction, since the simulated circuit is intended for implementation at later stages of design. Fixed propagation delays can be implemented in A/D and D/A converters, so the requirement for functional model are also not too restrictive.

Proposed solution for the PSS simulation of mixed signal RF circuits is shown in Fig. 2. Notice that there is no time step synchronization, which is a mayor difference from the setup shown in Fig. 1. Circuit simulator sees the wrapper Verilog-A module, which performs A/D and D/A conversion and provides interface to the digital simulator. Module inputs are sampled by a simple A/D converter with a small hysteresis to avoid oscillations during Newton-Raphson iterations, and digital outputs are evaluated at every time step.



```
module prbs4(clk, out);
  input clk;
  output out;
  reg [3:0] d=4'b0010;
  always (posedge clk)
    d <= {d[2:0], d[3]^d[2]};
  assign out = d[0];
endmodule
```

Fig. 2. Proposed solution for periodic steady state mixed-signal simulation

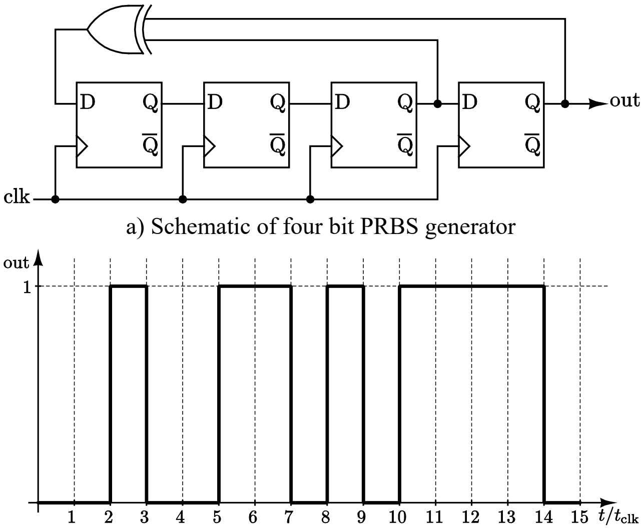
The process of generating the shared library and Verilog-A interface for a given Verilog module has been fully automated by a tool written in Python, called v2va (Verilog to Verilog-A). The tool parses the Verilog module hierarchy by invoking the Verilog-Perl scripts [9], generates the required C++ and Verilog-A wrappers from templates, invokes the Verilator, compiles the generated code and links all of the object files to a shared library, which can be used by Cadence Spectre circuit simulator.

At this point a question emerges: what happened to hidden states? A short answer would be: they are still present, but are now *very well* hidden states. A bit longer explanation is that the circuit simulator has no means of determining whether memory elements (hidden states) exist, since they are completely implemented in a shared library. As far as the circuit simulator is concerned, there are no hidden states, since it cannot see them. Implementation of memory elements outside of scope of analog solver also means that the circuit matrix size is independent of number of memory elements in digital block, and scales with the number of outputs instead.

Hiding hidden states from simulator has its pitfalls. Consider a four bit pseudo-random bit sequence (PRBS) generator, shown in Fig. 3a, implemented as maximum length linear feedback shift register with a characteristic polynomial:

$$f(x) = x^4 + x^3 + 1. \quad (1)$$

Generated bit sequence, shown in Fig. 3b, has a period of 15 clocks, so it is expected that the PSS simulation converges with a period of $T = 15t_{\mathrm{clk}}$, which can easily be confirmed by running a simulation.



a) Schematic of four bit PRBS generator

b) Output waveform

Fig. 3. Four bit PRBS generator example

However, simulation also converges for periods of

$$T_{\mathrm{false}} = \{1,3,4,7,9,14\} t_{\mathrm{clk}}, \quad (2)$$

because the periodicity condition $v(0) = v(T)$ holds for all signals visible to the simulator as well. False convergence does not arise due to a simulator bug, but due to extra states in digital model which are not visible. The same circuit with output ports at all flip-flop outputs does not exhibit false convergence, i.e. converges only for a period of 15 clocks.

To prevent false convergence, all memory element outputs should be visible to the simulator by connecting them to module output ports. This way the periodicity check is performed by the simulator, but might not be practical due to large number of signals.

To check whether false convergence has occurred, the PSS simulation results should be exported, and the digital simulation should be performed with exported stimuli. If false convergence has occurred, some of internal digital signals will have different values at the start and the end of simulation. This check can be automated with a few scripts.

IV. Periodic Steady State Mixed-Signal Simulation Example

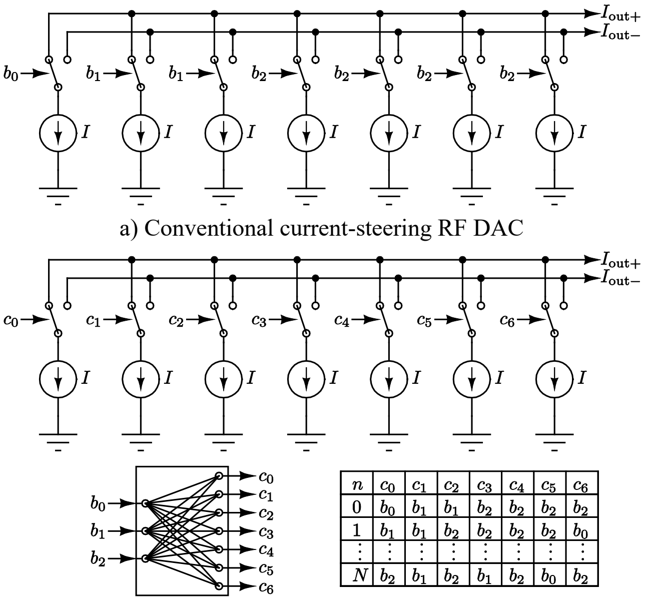
Advances in CMOS technology have resulted in expansion of digitally assisted RF circuits, offering unprecedented levels of performance. Direct RF sampling mixer [2] and dynamic element matching performance improved RF D/A converter [10] demonstrate the benefits of digitally assisted RF circuits. RF D/A converter will be used as an example for the PSS mixed-signal simulation demonstration.

Transistor matching requirements for a Nyquist rate current-steering digital to analog converters are set by the number of bits and expected yield [11]. As the number of bits increases, both unit transistor area and the number of transistor increase, becoming prohibitively large. Various calibration techniques [12][13] have been developed to relax the matching and area requirements. For a special case of RF D/A converter, dynamic performance, such as spurious-free dynamic range (SFDR), is of primary importance. This insight can be used to significantly reduce the chip area, as shown in [10].

Current-steering D/A converter output can be thought of as a sum of desired signal and spurious response generated by mismatch. Spurios response amplitude is determined by the mismatch current, while the number of spurs and their frequencies are determined by the repetition rate and pattern of data bit.

In a conventional binary weighted current-steering D/A converter data bit always controls the same current source, as shown in Fig. 4a. Any mismatch in current source current will produce a spurious response determined by data bit repetition rate and pattern. Spurious response can be reduced by increasing the transistor size to improve matching, or by employing an elaborate calibration scheme, which might require auxiliary circuits.

Randomized dynamic element matching current steering D/A converter principle of operation [10] is shown in Fig. 4b. Each data bit controls the number of current sources corresponding to its binary weight, but in contrast to conventional architecture, the current sources are dynamically assigned at each clock cycle by a predetermined algorithm, usually in a pseudo-random manner. This way the current source, and also its mismatch current, is not controlled by a specific data bit, but is pseudo-randomly assigned to all data bits. Pseudo-random assignment to data bits breaks the repetitive patterns and spreads the spectral components due to mismatch currents, effectively reducing spurs and increasing SFDR.



a) Conventional current-steering RF DAC

n	c_0	c_1	c_2	c_3	c_4	c_5	c_6
0	b_0	b_1	b_1	b_2	b_2	b_2	b_2
1	b_1	b_1	b_2	b_2	b_2	b_2	b_0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	b_2	b_1	b_2	b_1	b_2	b_0	b_2

b) Randomized dynamic element matching current-steering RF DAC

Fig. 4. Conventional and RDEM current-steering RF DACs

Randomized dynamic element matching technique significantly relaxes the current source matching requirements, leading to greatly reduced chip area, as demonstrated in [10]. The technique relies heavily on digital circuits, such as pseudo-random number generator and multiplexer matrix for data bit routing, making it an ideal candidate to demonstrate the effectiveness of proposed solution for PSS mixed-signal simulation.

A 10 bit randomized dynamic element matching RF D/A converter, similar to the one from [10] and shown in Fig. 5, has been designed in 65 nm CMOS for demonstration purposes. Eight MSB bits have been assigned to two four bit rotation-based binary weighted D/A converters, while the remaining two LSB bits have been assigned to conventional D/A converter. Data bits have been expanded at inputs of barrel shifters and a buffer, according to their binary weight.

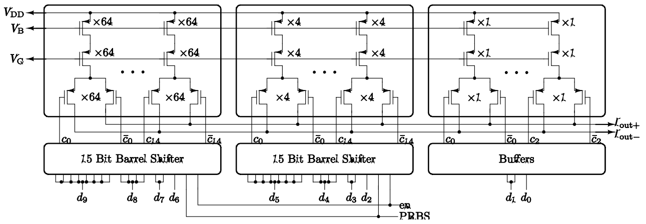


Fig. 5. Designed 10 bit randomized dynamic element matching RF D/A converter schematic

Current sources and current-steering differential pairs have been designed on a transistor level to allow Monte Carlo simulations with foundry-provided models, and to capture the spectral components originating from switch charge injection. Unit current source transistor area has been chosen for drain current mismatch standard deviation of 5%, which corresponds to 3 bit macthing. Transistor W/L was chosen for 250 mV overdrive at nominal 10 µA drain current.

Data multiplexers and 12 bit PRBS generator have been designed as Verilog modules and compiled with v2va tool. Each of four bit D/A converters' data randomizers have been designed as 15 bit four stage barrel shifters, so only right or left rotations of data bits are possible. Data randomizers have an enable input to control whether data bits are rotated or not. When enable bit is set to 0, the converter is a conventional D/A converter, while setting enable to 1 activates the randomized dynamic element matching. To avoid false convergence, all flip-flop outputs of PRBS generator are visible to the analog simulator.

For the purpose of testing the D/A converter, 10 bit digital look-up table sine generator was designed in Verilog. The period of 12 bit PRBS generator is 4095 clock cycles, and the sine generator was designed to have the same number of look-up table entries. To ensure signal coherence [14] and to reduce the impact of repetitions in quantization noise, which result in quantization noise spurs, the number of sine wave periods should be coprime to the total number of waveform samples. Since $4095 = 3^2 \cdot 5 \cdot 7 \cdot 13$ is a composite number, choosing the number of sine wave periods not divisible by 3, 5, 7 and 13 ensures the signal coherence. The digital sine generator was designed to have 127 periods in 4095 samples, which satisfies the coherence condition. Clock frequency was set to 1 GHz.

Although the generated input signal is a sine wave, output spectrum is expected to have significant spurs due to mismatch of transistors in current mirrors. To determine the spurious response due to mismatch, ten runs of Monte Carlo simulation were run. Worst case SFDR without randomized dynamic element matching was in the 7th iteration, is shown in Fig. 6. The 7th iteration simulation was repeated with randomized dynamic element matching, and the results are shown in Fig. 7.

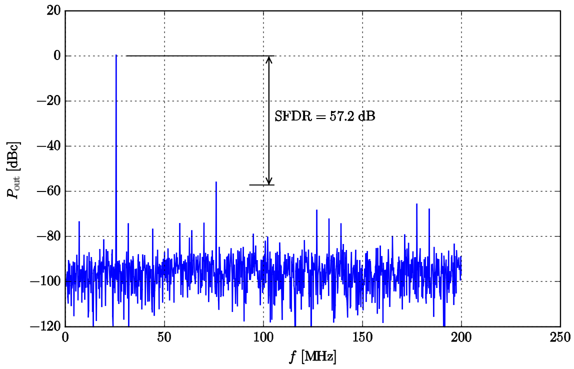


Fig. 6. Worst case SFDR without randomized dynamic element matching in ten Monte Carlo runs

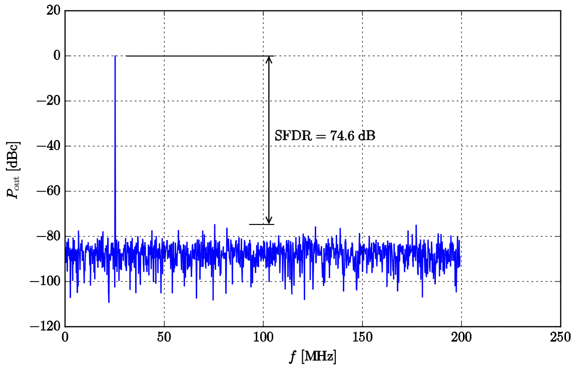


Fig. 7. SFDR improvement with randomized dynamic element matching

Without randomized dynamic element matching, SFDR of 57.2 dB can be achieved with chosen transistor size, which is not adequate for a 10 bit D/A converter. By enabling the randomized dynamic element matching, SFDR is improved to 74.6 dB – an improvement of 17.4 dB. SFDR of 74.6 dB is adequate for 10 bit D/A converter.

The importance of mixed-signal approach can be seen from the reduction in the number of transitors. Simulated circuit has approximately 150 transistors. A single PSS simulation lasts about 20 minutes, requiring 4 GB of memory. It is estimated that two four state 15 bit barrel shifters and a 12 bit PRBS generator would require 1500 transistors for implementation, leading to tenfold increase. Apart from sheer number of transistors, number of time points would be significantly increased, since all of the internal signal propagation would have to be simulated. This would render even a single PSS simulation infeasible.

V. Conclusion

Conventional mixed-signal simulators can only perform basic types of simulation, such as transient, AC and linear noise analysis. PSS-capable simulators do not support digital simulation, and cannot simulate circuits with hidden states. In this paper, we have proposed a solution based on compiling a digital design written in synthesizable subset of Verilog into a shared library with Verilator, and making a Verilog-A wrapper for simulator interface. The proposed solution might exhibit false convergence, which can be prevented and/or detected. The proposed PSS mixed-signal simulation flow has been demonstrated on dynamically matched RF D/A converter, where it was clearly shown that SFDR can be significantly improved by employing digitally assisted techniques.

References

[1] Open Verilog International, "*Verilog-A Language Reference Manual Version 1.0*," Aug. 1996.

[2] Muhammad, K., Staszewski, R. B. and Leipold, D. "*Digital RF processing: toward low-cost reconfigurable radios*," Communications Magazine, IEEE, vol. 43, pp. 105-113, 2005.

[3] Kundert, K., "Hidden State in SpectreRF," *The Designer's Guide (2003)*.

[4] Kundert, K., "*The Designer's Guide to Spice and Spectre®*," Springer Science & Business Media, 2006.

[5] Snyder, W.; Verilator, available at http://www.veripool.org/wiki/verilator

[6] Snyder, W., "*Verilator: Fast, Free, But for Me?*," Design Verification Club meeting, Bristol, Cambridge & Eindhoven, 2010.

[7] Snyder, W., "*Verilator: Open Simulation-Growing Up*," Design Verification Club meeting, Bristol, Cambridge & Eindhoven, 2013.

[8] Braend, D., Kruse, R., Xu, J., and Ruund, J. E., "*Atmel and the use of Verilator to create uC Device Models*," Atmel presentation

[9] Snyder, W.; Verilog-Perl, available at webpage http://www.veripool.org/wiki/verilog-perl

[10] Lin, W. T., Kuo, T. H., "*A Compact Dynamic-Performance-Improved Current-Steering DAC With Random Rotation-Based Binary-Weighted Selection*," IEEE Journal of Solid-State Circuits, Vol. 27, No. 2, pp. 444-453, 2012

[11] van den Bosch, A., Steyaert, M. S. J., and Sansen, W. "*An accurate yield model for CMOS current-steering D/A converters*," in Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), 2000, pp. 105–108.

[12] Bugeja, A. R., and Song, B. S., "*A self-trimming 14-b 100-MS/s CMOS DAC*," IEEE J. Solid-State Circuits, vol. 35, no. 12, pp. 1841–1852, Dec. 2000.

[13] Shen, D. L., Lai, Y. C., and Lee, T. C., "*A 10-bit binary-weighted DAC with digital background LMS calibration*," in Proc. IEEE Asian Solid-State Circuits Conf. (ASSCC), Nov. 2007, pp. 352–355.

[14] Lundberg, K. H., "Analog-*to-digital converter testing,*" Massachusetts Institute of Technology, 2002.