

Integrating Circuit Analyses for Assertion-based Verification of Programmable AMS Circuits

Dogan Ulus
Dept. of Electrical & Electronics Eng.
Bogazici University
Istanbul, Turkey
Email: dogan.ulus@boun.edu.tr

Alper Sen
Dept. of Computer Eng.
Bogazici University
Istanbul, Turkey
Email: alper.sen@boun.edu.tr

Faik Baskaya
Dept. of Electrical & Electronics Eng.
Bogazici University
Istanbul, Turkey
Email: faik.baskaya@boun.edu.tr

Abstract—Digitally-programmable analog circuits provide re-configurability and flexibility for next-generation electronic systems and modern electronic systems need such circuits more than ever. For verification of these circuits, the change in analog characteristics according to digital inputs should be monitored and checked to determine whether measured analog characteristics satisfy desired conditions in a unified analog mixed-signal (AMS) verification environment. Therefore, we integrate common analog circuit analyses into an assertion-based verification flow, and we verify time-varying analog characteristics of digitally-programmable AMS circuits. We use results of DC, AC and Fourier transform based analyses in our AMS assertion language, and monitor violations caused by any change in digital inputs. We show an application of our approach on a programmable low-pass filter circuit where cut-off frequency can be digitally controlled.

I. INTRODUCTION

There are growing demands for mixed-signal circuits especially for system-on-a-chip applications. However, a desire for more functionality in a more compact device has increased the level of complexity of these applications and made their verification harder than ever. Especially bugs that can occur at the boundaries of analog and digital blocks may remain hidden until the very end because analog and digital blocks are verified separately. To prevent such bugs and verify analog and digital blocks together, some proven techniques from digital domain have been extended toward analog domain in recent years.

Among these digital techniques, assertion-based verification (ABV) has gained popularity for its practicality and scalability. Formal description of AMS properties and automatic evaluation of simulation runs are two main benefits of ABV flow for mixed-signal designs. Assertion-based methodology reduces manual effort, increases reusability and leads to a productivity increase in AMS verification.

These digital techniques are built on top of the analog verification techniques such as circuit analyses. Assertions can check properties from time-domain simulation (transient analysis) results but these assertions are mostly safe operating area checks, that observe voltages and currents in design. For

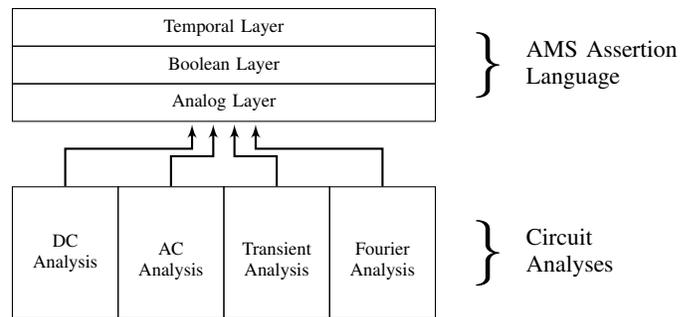


Fig. 1. General structure of an assertion-based AMS verification environment including circuit analyses

performance related properties, analog designers perform other types of analog analyses for their designs besides transient analysis. Therefore, assertions for other analog analyses and their integration into mixed-signal time-domain verification still remains a challenge.

In this paper, we integrate circuit analyses into assertion-based verification flow and we verify a specific class of mixed-signal circuits, that is digitally-programmable analog circuits, where analog circuit characteristics can be controlled by switching digital inputs. In Figure 1, we show our unified assertion-based verification flow that includes circuit analyses such as DC, AC, transient, Fourier transform based analyses as well as digital verification. In our approach, assertion-based flow can check and monitor results of circuit analyses whenever a digital event changes the state of the analog circuit.

In the next section, we briefly give a summary of assertion-based AMS verification from the literature. In Section III, we present the grammar of our assertion language integrating circuit analyses for assertion based verification flow. In Section IV, we explain DC, AC and Fourier transform based analyses and their integration into mixed-signal verification and we validate our approach on a programmable low-pass filter circuit in Section V.

II. RELATED WORK

Several formal or semi-formal approaches for analog mixed-signal (AMS) circuit verification are presented in [1] and [2]. Formal approaches like [3] and [4] use state-space discretization and symbolic computations for AMS verification. Alternatively, works proposing semi-formal methods use simulation traces to verify AMS properties. Expressive monitors and checkers are crucial for simulation-based approach, therefore several specification languages have been proposed to describe assertions addressing different aspects of AMS verification.

In most monitoring techniques, predicates over real-valued signals convert real-valued signals into Boolean signals and temporal properties are monitored and checked in simulation traces. Among them, *Signal Temporal Logic* (STL) [5], which extends the *Metric Interval Temporal Logic* (MITL) [6], is presented to monitor time-domain properties of continuous signals. The underlying idea and logic of STL is extended with many useful constructs such as auxiliary state machines in [7], frequency-domain operators in [8] and haloes in [9]. Similarly, specification languages in [10] and in [11] use assertions to verify AMS designs but they use a discrete-time notion in synchronization between analog and digital domains unlike a continuous time notion as in STL.

Expansion towards frequency domain is a natural and required step for more feature-rich assertion-based verification. Therefore, the authors of [10] include an FFT operator in their analog layer but it is limited to amplitude measurement only. In [8], the relation between time and frequency domain is discussed from AMS assertion perspective, and Time-Frequency Logic (TFL) is introduced. However, important tools for analog verification such as DC, AC and other FFT-based analyses are missing in current assertion languages. In this paper, we investigate the benefits of integration between circuit analyses and current assertion languages.

III. OUR AMS ASSERTION LANGUAGE

Analog mixed-signal (AMS) assertion languages consist of several abstraction layers and are influenced by *Property Specification Language* (PSL) [12]. Alongside *Boolean* and *Temporal* layers as in PSL, AMS assertion languages introduce an *Analog* layer to capture analog properties. In early AMS assertion languages, analog layer is only capable to process transient analysis results. However, other circuit analyses used in traditional analog verification is crucial to verify all aspects of analog design. Therefore, we extend our AMS assertion language to handle other types of circuit analyses such DC, AC and Fourier analysis.

We show the grammar of our AMS assertion language in Table I. The temporal and boolean operators have their usual semantics. *DC*, *AC*, *TRAN* and *FFT* operators denote corresponding operators. Note that *FFT* operator needs parameters such as *number of points* and *window function*, which we do not go into detail in Table I. *WF* operator denotes custom waveforms to define desired values of measured circuit properties. AC property operators, shown as \mathcal{A} , calculate AC properties like bandwidth *BW* from a given AC analysis.

TABLE I
OUR AMS ASSERTION LANGUAGE GRAMMAR

Temporal Layer	$TempExpr$	$::=$	$\odot BoolExpr$ $\odot TempExpr$ $TempExpr \bullet TempExpr$ $\neg TempExpr$
	$BoolExpr$	$::=$	$Signal = [ubound, lbound]$ $BoolExpr \bullet BoolExpr$ $\neg BoolExpr$
Boolean Layer	$Signal$	$::=$	$RawSignal \odot RawSignal$ $RawSignal$ $Const$
	$RawSignal$	$::=$	$dcExpr$ $acExpr$ $tranExpr$ $fftExpr$ $CustExpr$
Analog Layer	$dcExpr$	$::=$	$DC(Node)@{\ Events}$
	$acExpr$	$::=$	$\mathcal{A}(AC(Node)@{\ Events})$
	$tranExpr$	$::=$	$TRAN(Node)$
	$fftExpr$	$::=$	$\mathcal{F}(FFT(Signal, params)@{\ Events})$
	$CustExpr$	$::=$	$WF(Name)@{\ Events}$
	$Events$	$::=$	$Event$ $Events, Event$
		\odot Temporal Operators	<i>DC</i> DC Analysis Operator
		\bullet Binary Boolean Operators	<i>AC</i> AC Analysis Operator
		\neg Boolean Negation Operator	<i>TRAN</i> Transient Analysis Operator
		\odot Arithmetic Operators	<i>FFT</i> Fourier Analysis Operator
		\mathcal{A} AC Property Operator	<i>WF</i> Custom Waveform Operator
		\mathcal{F} Fourier Property Operator	

Similarly, FFT property operators, shown as \mathcal{F} , calculate FFT properties like total harmonic distortion *THD* from a given FFT analysis. An example property using our grammar is as follows:

$$Always(BW(AC(V(out))@{\ e_1}) = [10e6, 12e6])$$

where *out* denotes an output node of an analog filter. For this assertion, whenever an event e_1 occurs, bandwidth (*BW*) operator in the analog layer calculates the bandwidth from AC analysis results. Events can be defined as analog (such as a signal crossing a voltage value) or digital events (a change in corresponding digital signal) in our context. Then, bandwidth value is checked to see whether it is between 10MHz and 12MHz in Boolean layer, and *Always* operator in Temporal layer checks whether it is true for all simulation times.

IV. CIRCUIT ANALYSES

Circuit analyses are well-defined ways of collecting information (metrics) about analog designs. Because circuit analyses provide valuable metrics for verification, we see a great benefit to integrate them into assertion-based mixed-signal verification.

Among circuit analyses, DC operating point analysis determines the quiescent state of circuits. AC analysis extracts small-signal linear response of the circuit for a single frequency input around the DC operating point. Transient analysis computes time-domain response of the circuit by iteratively solving the algebraic differential system of equations. Fourier analysis, by using Fast Fourier Transform (FFT), returns power spectrum of time-domain signal so that we can see how much power resides in frequency components of that signal. We can analyze noise and linearity characteristics of analog circuits

from power spectra. In a mixed-signal design environment, digital verification techniques like assertions are still available on top of circuit analyses.

In this paper, we focus on DC, AC and Fourier analyses and their integration into AMS assertion languages. We use transient analysis, which is essentially time-domain simulation of analog circuits, as a bridge between DC, AC and Fourier analyses in the verification of programmable analog circuits. It means that we start a transient analysis and we pause transient analysis whenever a digital event changing the state of analog circuit occurs. Then, we check DC, AC or Fourier characteristics of the circuit for that time instant. This way, we can monitor and verify time-varying characteristics of programmable analog circuits. We now describe each of these circuit analyses in more detail.

A. DC Operating Point Analysis and DC Assertions

DC operating point analysis determines the quiescent state or stable initial condition of the circuit. The quiescent state is computed by the simulator with capacitances opened and inductances short-circuited. It provides operating point information before an small-signal (AC) analysis or a transient analysis so it is the starting point in the analog design flow.

In regular flow of analog design, designers make assumptions and choices for DC values inside their analog blocks. Assertions capture these assumptions and design choices and they report if there is undesired situation. DC characteristics of analog circuits include DC voltage levels of circuit nodes, DC current values for circuit branches as well as operating modes and conditions for circuit devices.

Related analog circuit specifications that can be extracted via DC analysis include offset voltage values, offset voltage drifts, bias current values, current drifts and operation modes of transistor devices. For example, if intended DC voltage level for the output node is between 0.85V and 0.95V for an amplifier design, we can capture this specification in following formula.

$$DC(V(out)) = [0.85, 0.95]$$

For time-invariant analog circuit blocks, DC analysis is performed at the start of simulation and it is considered as valid for all simulation. Therefore, single DC analysis is sufficient to verify DC characteristics for time-invariant systems. However, in case of circuits having time-varying DC characteristics such as adaptive or digitally controlled analog circuits, it is beneficial to integrate DC analysis checks into time-domain simulation (transient analysis). This way, the interaction between digital and analog domains can be investigated in a single simulation environment. To ensure desired DC characteristics during such a mixed-signal simulation, we integrate DC analysis checks into AMS assertions. For example, we want to check DC voltage level for the output node of an amplifier with digitally-adjustable transconductance. Because transconductance adjustment by digital events can shift DC voltage levels in the analog circuit, output DC

voltage level should be monitored and checked during time-domain simulation via the following assertion.

$$Always(DC(V(out))@ \{e_1\} = [0.85, 0.95])$$

where e_1 is denoted as a digital event such as a change in digital control inputs. This assertion monitors DC value of output node and checks if it satisfies the desired condition when event e_1 occurs anytime during simulation.

B. AC Analysis and AC Assertions

Traditionally, designers analyze analog circuits by exciting them with a single-frequency input signal, and they measure the output signal to see how the magnitude and phase of the input signal is changed by the circuit. In general, only linear analog circuits can be analyzed using this technique; therefore, a nonlinear circuit should be linearized by assuming input signals are small enough. This technique, called as small-signal analysis or AC analysis, is a common and useful procedure when analyzing frequency response of analog circuits.

Related analog circuit specifications that can be extracted via AC analysis include DC gain, bandwidth, phase margin, gain margin and roll-off slopes. In Figure 2, we show an example AC analysis plot where we can see measured AC metrics for the output node of a low-pass filter circuit. We can write assertions to check whether measured metrics satisfy the specifications in assertion-based verification. For example, if desired value for DC gain is between 0dB and -0.5dB for a low-pass filter design, this specification is captured as:

$$dcgain(AC(V(out))) = [0, -0.5]$$

In linear time-invariant (LTI) analog blocks, AC analysis is performed at the start of simulation after DC analysis and it is considered as valid for all simulation. Therefore, single AC analysis is sufficient to verify AC characteristics for LTI

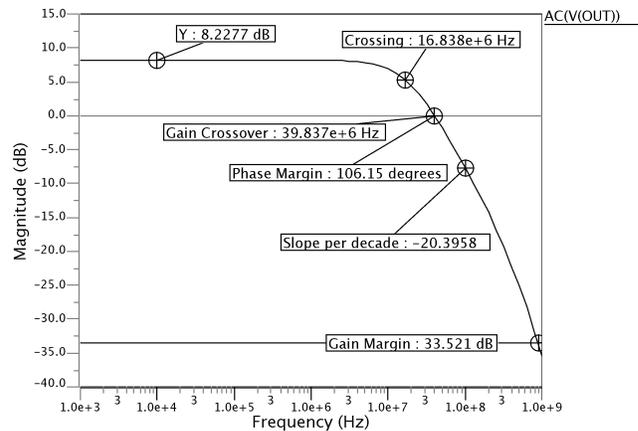


Fig. 2. An example AC analysis plot and measured AC specifications

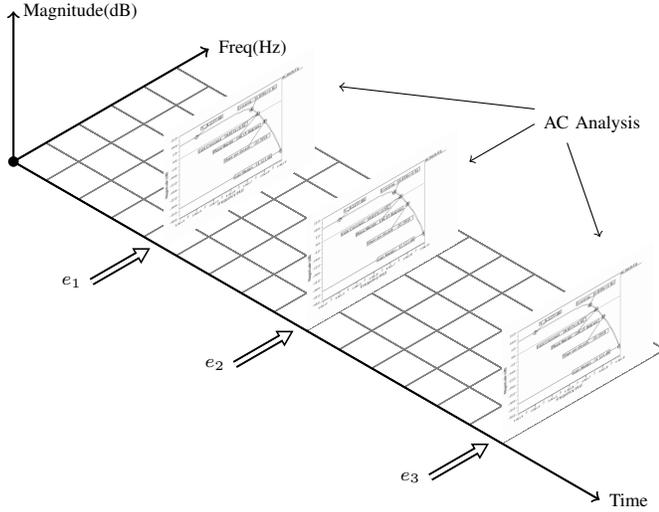


Fig. 3. Monitoring time-varying AC characteristics in the middle of time-domain circuit simulation. A new AC analysis is performed when an event changes the state of the circuit.

systems. However, in case of circuits having time-varying AC characteristics such as adaptive or digitally controlled analog circuits, it is beneficial to integrate AC analysis checks into time-domain simulation (transient analysis) as we did for DC assertions.

For example, if we want to check DC gain value for a low-pass filter with digitally-adjustable cut-off frequency, then we write the following assertion to capture DC gain property.

$$\text{Always}(\text{dcgain}(\text{AC}(\text{V(out)})@ \{e_1, e_2, e_3\}) = [0, -0.5])$$

where e_1 , e_2 and e_3 are denoted as discrete events such as changes in digital control inputs. In Figure 3, we illustrate how AC analysis is performed in the middle of a time-domain simulation. During time-domain simulation, AC analysis is performed for time instants when events e_1 , e_2 , e_3 occur. Then, our assertion monitors DC gain at the output node from AC analysis results and checks if it always satisfies the desired condition when these events occur.

An important limitation for us: Although current simulators can perform AC analysis in the middle of transient analysis, they use current node voltages for AC analysis instead of performing true DC analysis. This limitation makes each AC analysis which are performed at exact event time wrong. Therefore, we solve this problem practically by delaying corresponding event times (until DC voltages are stabilized) for AC analyses. With simulators allowing event-driven DC and AC analysis, analysis and verification of time-varying analog circuits are easier and more comfortable.

C. Fourier Analysis for Noise and Linearity

Noise and linearity characteristics of analog circuits determine the range of useful signals that the circuit processes as intended. The smallest value of signals that can occur is limited

by the noise floor of the circuit whereas the the largest value of signals is limited by nonlinearity of the circuit. Therefore, we use Fast Fourier Transform (FFT) based analyses to verify dynamic range of analog circuits in practice.

Dynamic range metrics include total harmonic distortion (THD), signal-to-noise and distortion ratio (SNDR) and spurious free dynamic range (SFDR). We can extract these metrics by performing a FFT to get the power spectrum of time-domain simulation results if the circuit is excited with a single-frequency input. On the other hand, we should consider all factors to compute a healthy FFT such as number of points, windowing choice and sampling frequency during these analyses.

In Figure 4, we illustrate an example power spectrum plot, which shows the power of each frequency component along frequency axis. The biggest peak implies the fundamental frequency (13MHz in this case), where most of the power is concentrated, and we can see noise and distortion components in other frequencies.

1) *Total Harmonic Distortion*: The total harmonic distortion (THD) of a signal is the ratio of the total power of all second and higher harmonic components to the power of the fundamental harmonic (first harmonic) for that signal. Because a nonlinear system produces second, third and higher-order distortion components at the harmonics of the input (fundamental) frequency, when excited with a sinusoidal source, it is used as measurement for the nonlinearity of a system. THD metric is calculated by the formula below, and is expressed in gain (dB) or percentage.

$$\text{THD} = 10 \log \frac{H_{D2}^2 + H_{D3}^2 + \dots + H_{Dn}^2}{H_{D1}^2}$$

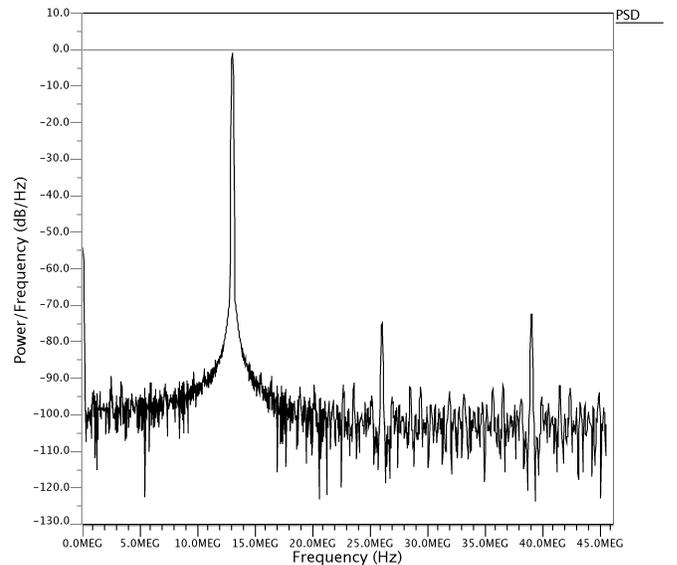


Fig. 4. Power Spectrum Example

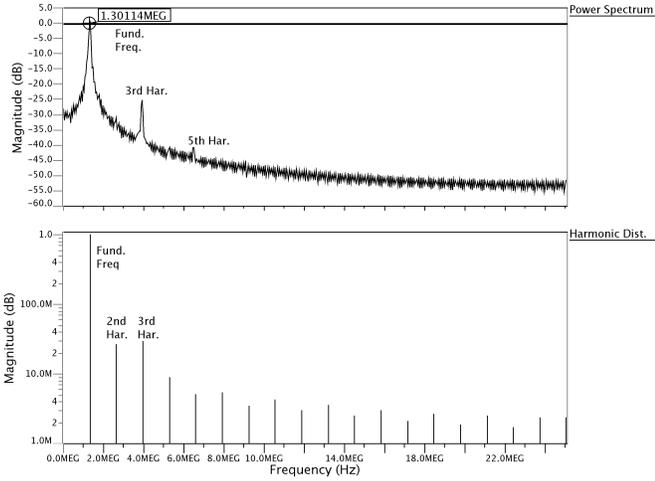


Fig. 5. THD Example

where H_{D1}^2 , H_{D2}^2 , H_{D3}^2 and H_{Dn}^2 represent power value of first-, second-, third- and n-th order harmonics, respectively.

In Figure 5, we illustrate an example THD analysis of an analog circuit, which is excited by an input frequency of 1.3 MHz. In the normalized power spectrum of output signal, we can see the biggest peak at 1.3 MHz, which is the fundamental frequency, and a few smaller peaks at the integer-multiples of fundamental frequency, which are called harmonics. Then, THD operator calculates the ratio of the power of fundamental frequency and the power of all other harmonics. We write an assertion to monitor time-varying THD characteristics of any weakly nonlinear analog circuit as follows:

$$\text{Always}(\text{THD}(\text{FFT}(\text{V}(\text{out}))@ \{e_1\}) = [0, 2.5])$$

This assertion checks if calculated THD value always satisfies the desired specification at the time instants when event e_1 occurs.

2) *Signal-to-Noise and Distortion Ratio*: Another widely-used metric for noise and linearity specification is Signal-to-Noise-and-Distortion Ratio (SNDR). It is the ratio of signal power to power sum of all other spectral components, and expressed in dB. SNDR is a good indicator about system's performance because it takes both distortion and noise components into account. SNDR is calculated by using the formula below:

$$\text{SNDR} = 10 \log \frac{P_{\text{signal}}}{P_{\text{noise}} + P_{\text{distortion}}}$$

where P_{signal} , P_{noise} and $P_{\text{distortion}}$ denote the value of power of signal, noise and distortion components. We write

an assertion to monitor time-varying SNDR characteristics of any weakly nonlinear analog circuit as follows:

$$\text{Always}(\text{SNDR}(\text{FFT}(\text{V}(\text{out}))@ \{e_1\}) = [30, \text{Inf}])$$

This assertion checks if calculated SNDR value always satisfies the desired specification at the time instants when event e_1 occurs.

3) *Spurious Free Dynamic Range*: Spurious-free dynamic range (SFDR) is the ratio of the input signal to the peak spurious component. Spurs can occur at the harmonics of fundamental frequency because of nonlinearity or at other frequency values because of mismatches in the circuit. In Figure 6, we illustrate an example SFDR analysis of an analog circuit. In the normalized power spectrum, we see that the ratio of the input signal power and the power of biggest peak component (third harmonic) is 23 dB.

We write an assertion to monitor time-varying SFDR characteristics of any weakly nonlinear analog circuit as follows:

$$\text{Always}(\text{SFDR}(\text{FFT}(\text{V}(\text{out}))@ \{e_1\}) = [20, \text{Inf}])$$

This assertion checks if calculated SFDR value always satisfies the desired specification at the time instants when the event e_1 occurred.

V. CASE STUDY

In a typical receiver system used in communication applications, three basic operations are performed: amplification, filtering and data conversion. Programmable gain amplifiers (PGAs), low-pass filters (LPFs), analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) are designed to implement these operations in actual circuits. In next-generation applications, these circuits should be very flexible

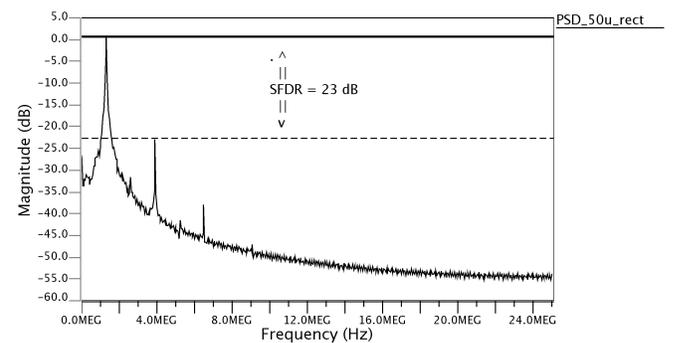


Fig. 6. SFDR Example

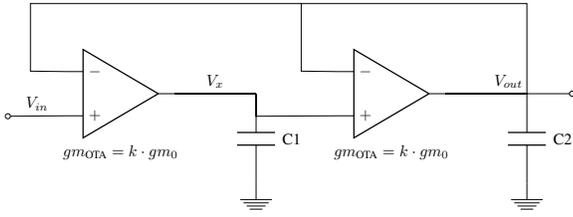


Fig. 7. The programmable low-pass filter circuit

and capable of adapting their performance to different standard requirements with reduced power consumption [13]. To achieve this, we need to design and verify analog circuits with digital control of biasing, gain, circuit- and stage-level reconfiguration, block power down/up. Therefore, we present a digitally-programmable continuous-time Gm-C low-pass filter design to show how our assertion based verification for mixed-signal systems using circuit analyses can be used.

Designers can implement filter designs in both digital and analog domains. Domain selection includes many design trade-offs and challenges. Although filtering at digital domain is preferred over analog domain, because of digital domain's robustness and scalability, the overwhelming requirements for following data conversion step make programmable analog filters attractive for new generation mixed-signal applications. Among two common programmable analog filter methods, active-RC and Gm-C circuit techniques, Gm-C technique is preferred for high-frequency and low-power applications.

To demonstrate our approach, we have designed a digitally-programmable low-pass Gm-C filter by using architectures and ideas presented in [14] and [15] for 0.18um technology and we simulated the circuit with Eldo SPICE simulator. In following figures, SPICE notation is used for numbers in plots. (M stands for milli, and MEG for mega in SPICE.) Schematic of the filter circuit can be seen in Figure 7, where amplifier blocks in the circuit are classified as operational transconductance amplifier (OTA). Cut-off frequency for this circuit is determined using the following equation:

$$f_c = k \times \frac{gm_0}{2\pi\sqrt{C_1 \cdot C_2}} \times 0.6412 \quad (1)$$

where f_c denotes cut-off frequency of the filter, g_m denotes transconductance value of amplifiers, C_1 , C_2 denote capacitor values in the circuit and k value provides programmability for this filter. We can change cut-off frequency of the low-pass filter during operation by changing k .

As in [15], we achieve such programmability in actual circuit by connecting three OTAs parallel where OTAs have binary weighted transconductance values, gm_0 , $2gm_0$ and $4gm_0$, respectively. Each OTA has a dedicated digital control bit to drive transistor switches inside the OTA circuit and this way we can turn it on or off. Because transconductance values of parallel OTAs are added up if OTAs are turned-on, we can control overall transconductance by adjusting three control bits of OTAs. The k value represents binary-coded decimal 3-bit

digital input and it is connected to control bits of OTAs for transconductance adjustment.

Note that $k = 0$, thus binary "000", is an invalid value for low-pass filter circuit according to Equation 1. Therefore, before starting to check circuit characteristics, we write an assertion to ensure that k is never equal to zero. We captured this property as follows:

$$\text{Always}(k[0] \vee k[1] \vee k[2]) \quad (\text{A0})$$

where $k[n]$ denotes Boolean value of n th bit binary-coded decimal k . This assertion warns if all bits of k is logic zero, thus $k = 0$.

In verification of programmable analog filters, we should check DC operating points of different states. The simulator performs a DC analysis and computes DC levels of all nodes. Then, we check if these values satisfy desired conditions for all time instants. However, changing 3-bit digital k value can disrupt DC levels in the circuit and cause a shift in DC level of the output node. On the other hand, ensuring that DC levels remains in a specified range is important for robustness of a system time-varying characteristics. Therefore, the simulator performs a DC analysis whenever a change in k occurs and we check DC levels for each change during simulation. In Figure 8, we illustrate the change in DC level of the output node based on time and varying k . If we want to keep the value of output DC level of the filter between 880mV and 920mV, we write this property:

$$\text{Always}(DC(V(\text{out}))@ \{e_k\} = [880\text{m}, 920\text{m}]) \quad (\text{A1})$$

where e_k denotes an event for a change in k determined by digital inputs. In Figure 8, we can see that varying DC level for output node is always between specified values so the assertion is satisfied. However, a stricter requirement for output DC level such as [895m, 905m] would fail for this circuit. We annotate the results of these checks in the form of boolean signals in Figure 8.

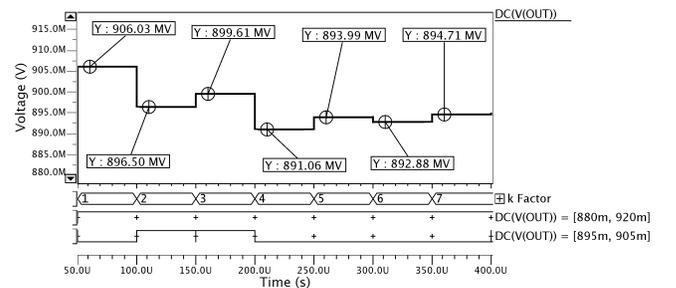


Fig. 8. Monitoring time-varying DC level of the filter output according to k .

As the next step, we verify time-varying AC characteristics of the filter. Design specification states that the cutoff frequency of the filter should be changed linearly depending on the value of k factor. According to Equation 1, cutoff frequency of the filter should be equal to $k \times 3$ MHz (eg. 3 MHz if $k=1$, 12 MHz if $k=4$) in the ideal case if we select circuit parameters, gm_0 , C_1 and C_2 as $60\mu A/V$, 4pF and 1pF, respectively. For monitoring cutoff frequency, we performed an AC analysis whenever k changed. We captured this property as follows:

$$\text{Always}(BW(AC(V(out)))@ \{e_k\} - WF(Id) = [-0.3e6, 0.3e6]) \quad (A2)$$

where $WF(Id)$ denotes a custom waveform (reference waveform) for desired values of cutoff frequency, and we check if the error value between actual values and desired values resides in a margin of tolerance. For each change in k , AC analysis returns a frequency response plot, that can be seen in Figure 9 as a combined plot. Bandwidth operator (BW) computes cutoff frequency from each AC analysis, and we plot cutoff frequency value of the filter versus time plot in Figure 10.

In Figure 10, we see that cutoff frequency values are not always in specified range so the assertion we wrote for cutoff frequency monitoring fails. According to Equation 1, incorrect transconductance values of operational transconductance amplifiers (OTAs) can lead to this erroneous behavior. Therefore, we write the following assertions to check whether transconductance values of both OTAs in the filter circuit satisfy desired values.

$$\text{Always}(GM(AC(V(x)), AC(V(in)), AC(V(out)))@ \{e_k\} - WF(Id) = [-5e6, 5e6]) \quad (A3)$$

$$\text{Always}(GM(AC(V(out)), AC(V(x)), AC(V(out)))@ \{e_k\} - WF(Id) = [-5e6, 5e6]) \quad (A4)$$

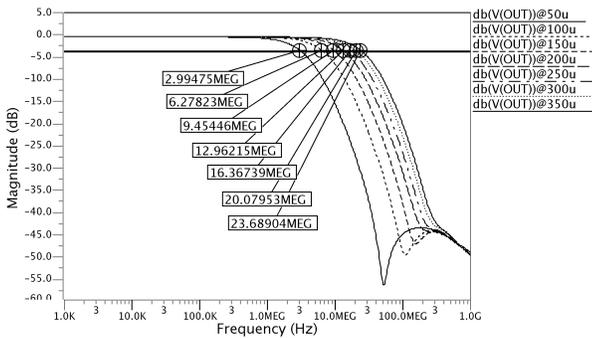


Fig. 9. AC analysis results obtained during time-domain simulation whenever k changes. We annotate the plots with calculated bandwidth values.

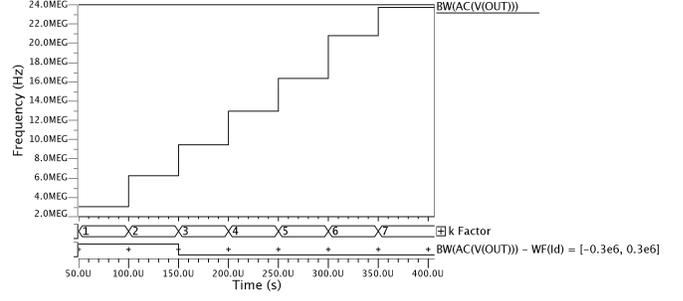


Fig. 10. Monitoring time-varying cutoff frequency value of the filter according to k factor.

where GM operator calculates transconductance value using AC analysis results of input and output nodes of OTAs, and $WF(Id)$ denotes a custom waveform for desired values of transconductance. This way we can trace the error up to the smallest circuit block and we can say which block is not working properly. Note that tracing error is an important benefit of our verification flow. Although writing assertions in hierarchical manner still requires design expertise, the checking process is now automatic, standardized and less error-prone.

Ultimately, a filter is designed to implement a linear operation however actual implementations of filter circuits (or any other circuit that implements a linear operation) are slightly nonlinear because of nonlinearity of transistor devices. FFT-based THD and SNDR metrics are used to measure the amount of nonlinearity of designs from transient simulation results. We captured THD and SNDR specifications as follows:

$$\text{Always}(THD(FFT(V(out)))@ \{e_k\} = [0, 3]) \quad (A5)$$

$$\text{Always}(SNDR(FFT(V(out)))@ \{e_k\} = [30, Inf]) \quad (A6)$$

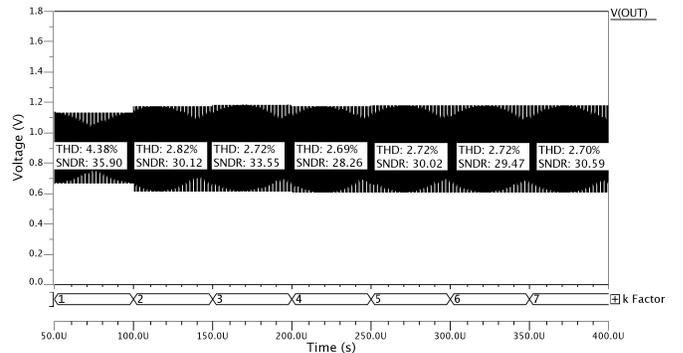


Fig. 11. Monitoring time-varying THD and SNDR values of the filter according to k .

In Figure 11, we plot the output node of the filter and monitor THD and SNDR properties for our filter design according to the change in k and we annotate calculated THD and SNDR values on the plot. We see that THD and SNDR values are not always in specified range so THD and SNDR specifications are not satisfied for all k . Unsatisfied linearity specifications usually do not have easy fixes and designers may require a change in circuit topology to achieve desired linearity. However, the same assertions can be used for several topologies; therefore, verification effort across different topologies is reduced compared to manual verification.

VI. CONCLUSION

Any simulation-based AMS verification methodology would be incomplete without an integrated support for circuit analyses. Therefore, we integrated circuit analyses such as DC, AC and Fourier analyses into assertion based verification flow, and proposed an AMS language including support for these analyses as well as transient analysis. This is a required step for a more unified and expressive environment for assertion based AMS verification. We performed our experiments on a specific class of analog circuits, digitally-programmable analog circuits whose analog characteristics can change in time. In future, we will analyze the applicability of our approach for other cases that include time-varying characteristics such as analog circuits with aging transistors.

ACKNOWLEDGMENT

This work was supported in part by Semiconductor Research Corporation under task 2082.001, Marie Curie European Reintegration Grant within the 7th European Community Framework Programme, and the Turkish Academy of Sciences.

REFERENCES

- [1] E. Barke, D. Grabowski, H. Graeb, L. Hedrich, S. Heinen, R. Popp, S. Steinhorst, and Y. Wang, "Formal approaches to analog circuit verification," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2009, pp. 724–729.
- [2] M. H. Zaki, S. Tahar, and G. Bois, "Review: Formal verification of analog and mixed signal designs: A survey," *Microelectron. J.*, pp. 1395–1404, 2008.
- [3] S. Steinhorst and L. Hedrich, "Analog assertion-based verification on partial state space representations using ASL," in *Proceedings of the Forum on Specification and Design Languages (FDL)*, 2012, pp. 98–104.
- [4] G. Al-Sammane, M. Zaki, and S. Tahar, "A symbolic methodology for the verification of analog and mixed signal designs," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2007, pp. 249–254.
- [5] O. Maler and D. Ničković, "Monitoring properties of analog and mixed-signal circuits," *International Journal on Software Tools for Technology Transfer*, pp. 1–22, 2012.
- [6] R. Alur, T. Feder, and T. Henzinger, "The benefits of relaxing punctuality," *Journal of the ACM (JACM)*, vol. 43, no. 1, pp. 116–146, 1996.
- [7] S. Mukherjee, P. Dasgupta, and S. Mukhopadhyay, "Auxiliary specifications for context-sensitive monitoring of ams assertions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 10, pp. 1446–1457, 2011.
- [8] A. Donze, O. Maler, E. Bartocci, D. Nickovic, R. Grosu, and S. Smolka, "On temporal logic and signal processing," in *Proceedings of the Symposium on Automated Technology for Verification and Analysis (ATVA)*, 2012, pp. 92–106.
- [9] D. Ulus and A. Sen, "Using haloes in mixed-signal assertion based verification," in *Proceedings of the Workshop on High Level Design Validation and Test (HLDVT)*, 2012, pp. 49–55.
- [10] S. Lämmermann, J. Ruf, T. Kropf, W. Rosenstiel, A. Viehl, A. Jesser, and L. Hedrich, "Towards assertion-based verification of heterogeneous system designs," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2010, pp. 1171–1176.
- [11] R. Mukhopadhyay, S. K. Panda, P. Dasgupta, and J. Gough, "Instrumenting AMS assertion verification on commercial platforms," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 14, no. 2, 2009.
- [12] H. Foster, E. Marschner, and Y. Wolfsthal, "IEEE 1850 PSL: The next generation," in *Proceedings of Design and Verification Conference and Exhibition (DVCON)*, 2005.
- [13] J. M. de la Rosa, R. Castro-Lopez, A. Morgado, E. C. Becerra-Alvarez, R. del Rio, F. V. Fernandez, and B. Perez-Verdu, "Adaptive CMOS analog circuits for 4G mobile terminals –review and state-of-the-art survey," *Microelectronics Journal*, vol. 40, no. 1, pp. 156 – 176, 2009.
- [14] R. L. Geiger and E. Sanchez-Sinencio, "Active filter design using operational transconductance amplifiers: A tutorial," *Circuits and Devices Magazine, IEEE*, vol. 1, no. 2, pp. 20–32, 1985.
- [15] S. Pavan, Y. P. Tsvividis, and K. Nagaraj, "Widely programmable high-frequency continuous-time filters in digital cmos technology," *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 4, pp. 503–511, 2000.