True Event-Driven Simulation of Analog/Mixed-Signal Behaviors in SystemVerilog: A Decision-Feedback Equalizing (DFE) Receiver Example

Ji-Eun Jang[1], Myeong-Jae Park[1], Dongyun Lee[2], Jaeha Kim[1]
[1]Inter-university Semiconductor Research Center, Seoul National University, Seoul, Korea
[2]Synerchip, Inc., Sunnyvale, CA, USA

***Abstract*- This paper presents a true event-driven simulation methodology for analog/mixed signal systems. To avoid generation of new output events without an input event, analog waveforms are expressed as a set of parameter values for an analytical basis function, $c \cdot t^{m-1} e^{-at} \cdot u(t)$. Also, the s-domain analysis enables an algebraic computation of the output event without involving time-step integration. The proposed methodology implemented in SystemVerilog is demonstrated with a decision-feedback equalizer (DFE) example. The experimental results show that both the speed and accuracy of the simulation depend very weakly on the time step resolution, supporting that a true event-driven simulation is realized.**

I. INTRODUCTION

As an increasing number of analog systems such as high-speed interfaces, RF transceivers, and DC-DC converters employ digital logic to overcome various shortcomings in analog, methodologies for fast, systematic verification of mixed-signal systems have become a necessity. However, one of the obstacles is the lack of mixed-signal behavioral simulators that can operate in a truly event-driven fashion, whose execution time does not scale with the system size but only with the amount of activities in the system. The key difficulties are two-fold: expressing continuous, analog signal waveforms in a way amenable to event-driven simulations; and computing a circuit's response to an analog signal without involving time-integration. While addressing these challenges, this paper presents a true event-driven methodology for simulating analog/mixed-signal behavioral models on a digital logic simulator such as SystemVerilog.

Most analog simulators express the signal waveforms as a sequence of time-value pairs and that is the first factor hindering true event-driven simulation. The most common way is to express a continuous-time waveform as a collection of piecewise linear (PWL) segments. With this waveform expression, however, the simulator must keep generating new events to model an output of an analog block that continuously changes even without any further changes in the input signal. For instance, considering a first-order low-pass filter that receives a step input, the output is an exponentially decaying waveform, which would take many PWL segments to properly express the waveform with small errors.

The second factor hindering true event-driven simulation is that most analog simulators compute the response of an analog system via time-integration. For instance, the most prevailing way of computing the response of a continuous-time dynamical system is to numerically solve an ordinary differential equation (ODE), e.g. using forward-Euler, backward-Euler, or trapezoidal integration methods [1,2]. Since the nature of these time-integration methods is that both the accuracy and numerical stability improve with the finer time steps, it is not sufficient to compute the system response only at the time instants of the input change events. Therefore, the simulation cost is likely to increase with the number of blocks that must be simulated via time-step integration. While there have been some recent efforts to simulate analog behaviors without using time-integration, for example, directly calculating the period of an oscillator [3] or looking up a pre-computed table for the system's output [4], they share the limitations that they are not generally applicable to arbitrary systems or input types.

This paper proposes a true event-driven methodology for simulating analog/mixed-signal behavioral models that can mitigate these two hindering factors. First, an analog waveform is expressed as a sequence of events, each updating a set of parameter values for an analytical equation that describes the waveform. For instance, when simulating the first-order filter's response to a step input, it is known that the response $y(t)$ can be in general described in an analytical form of $y(t) = y_0 \cdot exp(-t/\tau)$, as long as the input always changes in a step. Then, it is sufficient to update the coefficient y_0 and time constant τ only once at the time of the input step event. Second, this analytical expression of the output waveform can be derived directly without involving time-integration, provided that both the input signal and the system transfer function can be expressed in fractions of polynomials in s-domain after Laplace transformation. While it may sound like a restriction, the s-domain form can fully express a wide range of common analog signals, including steps, ramps, sinusoids, and exponentials.

The proposed methodology is an extension of our previous work [5] which demonstrated the primitive concept but used multiple different analytical basis functions for different types of analog signals (e.g. PWL, exponentials, PLL loop filter output). In addition, it could only compute the response of a linear system to a PWL input waveform. This work presents a single unified analytical basis function that can express all the above-mentioned signal types as well as their resulting system responses.

978-1-4673-1556-2/12/$31.00@2012 IEEE

II. True Event-Driven Simulation of Analog Behavioral Models

The basic problem of modeling a continuous system in an event-driven way is that the output signal can continuously change without any input change. To mitigate this problem, the proposed modeling methodology represents signals with a combination of continuous-time functions, not with a discrete sequence of time-value pairs. While nearly all circuit simulators adopt the latter method and produce a piece-wise linear (PWL) output waveform by connecting the discrete time-value points, it is an inefficient way to describe continuous-time signals in an event-driven simulation. For example, when a signal exponentially converges to a value A, as shown in Fig. 1, a PWL representation needs to keep generating a sequence of events to depict the signal within a certain error tolerance (Fig. 1(a)). If the output signal is represented by a combination of one step function and one exponential function (Fig. 1(b)), this representation can capture an exact output signal with only one event.

To express a broad range of signal types encountered in analog/mixed-signal circuits, we chose a basis function of $c \cdot t^{m-1} e^{-at} u(t)$, where $u(t)$ is a step function that takes the value of 1 for $t \geq 0$ and 0 otherwise. This basis function can express all the stimulus waveforms that mixed-signal designers commonly use for design verification. For instance, the basis function can express a step when m=1 and a=0, and a ramp when m=2 and a=0. Sinusoidal signals and their modulated signals can also be represented when a and c take complex values. An s-domain equivalent of $c \cdot t^{m-1} e^{-at} u(t)$ is

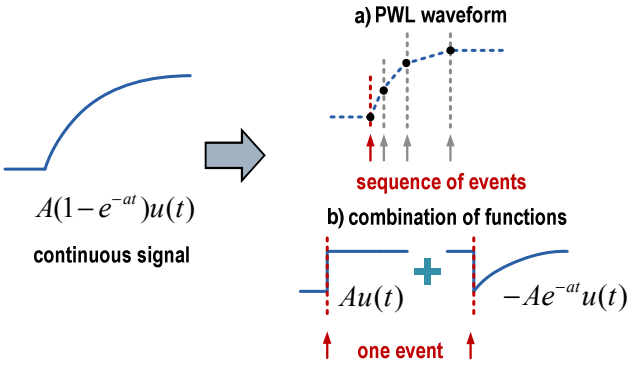

Figure 1. a) Piece-wise linear representation of continuous signal and b) proposed representation with a combination of functions

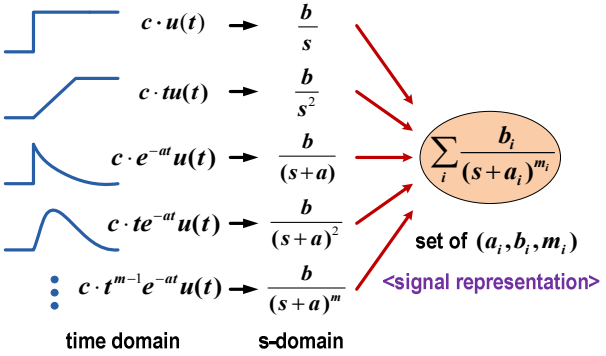

Figure 2. Basis function and its s-domain representation

$b/(s+a)^m$, and the proposed representation utilize these three parameters to depict this basis function: (a, b, m) as described in Fig. 2. If the signal is a combination of multiple basis functions, then the signal is represented as a set of parameter tuples: $\{(a_i,b_i,m_i)'s\}$.

The advantage of the aforementioned basis is that its time-domain expression can be easily transformed into Laplace s-domain, which enables the direct calculation of the output signal provided that the transfer functions of continuous blocks are also expressed in s-domain. For example, if a step is given as a input to a low-pass filter with one dominant pole at ω_p, the output signal expressed in s-domain is simply a product of 1/s and $1/(1+s/\omega_p)$. After partial fraction expansion, the expression becomes:

$$\frac{1}{s} \cdot \frac{1}{1+s/\omega_p} = \frac{1}{s} + \frac{-1}{s+\omega_p}. \quad (1)$$

According to our notation, the parameter sets for the input and output signals are $\{(0,1,1)\}$ and $\{(0,1,1), (\omega_p,-1,1)\}$, respectively, and this output corresponds to $(1-e^{-\omega_p t})u(t)$ in time domain. Even with more complicated transfer functions of which a numerator and a denominator are in the form of polynomials, partial fraction expansion can decompose them into a sum of rational functions with first-order denominators. Therefore, the process of calculating the output is nothing but summing the products between the input and each decomposed rational function of the transfer function. In general, the output can be calculated as in Eq. (2), where the input signal and transfer function are represented by the proposed signal representation.

$$\sum_i \frac{b_i}{(s+a_i)^{m_i}} \times \sum_j \frac{q_j}{(s+p_j)^{n_j}} = \sum_{i,j} \frac{b_i}{(s+a_i)^{m_i}} \times \frac{q_j}{(s+p_j)^{n_j}} \quad (2)$$

$$= \sum_{i,j} \left(\sum_{k=1}^{m_i} \frac{c_k}{(s+a_i)^k} + \sum_{l=1}^{n_j} \frac{d_l}{(s+p_j)^l} \right)$$

$$\text{where, } c_k = \begin{cases} \dfrac{b_i \cdot q_j}{(p_j - a_i)^{n_j}}, & for\ k = m_i \\ \dfrac{b_i \cdot q_j}{(p_j - a_i)^{m_i+n_j-k}} \cdot \dfrac{(-1)^{m_i-k}}{(m_i-k)!} \cdot \prod_{z=i}^{m_i-1} (m_i + n_j - 1 - z), & for\ k = m_i - 1, \cdots, 1 \end{cases}$$

$$d_l = \begin{cases} \dfrac{b_i \cdot q_j}{(a_i - p_j)^{m_i}}, & for\ l = n_j \\ \dfrac{b_i \cdot q_j}{(a_i - p_j)^{n_j+m_i-l}} \cdot \dfrac{(-1)^{n_j-l}}{(n_j-l)!} \cdot \prod_{z=i}^{n_j-1} (n_j + m_i - 1 - z), & for\ l = n_j - 1, \cdots, 1 \end{cases}$$

Since its final result also has the identical form with the proposed representation, the output of one block can be readily used as an input to the following blocks and this property allows multiple continuous blocks to be cascaded.

III. Implementation In SystemVerilog

This section discusses implementation issues in realizing the above-mentioned event-simulation methodology in System Verilog. SystemVerilog was chosen as our simulation platform as it can serve as a true event-driven simulation engine that seamlessly integrates analog and digital models

together. Also, one advantage of SystemVerilog compared to other HDL standards is that it offers a composite data type *struct* and also allows its instance variables to be passed across the port boundaries. Hence, the set of multiple parameters can be exchanged between the block models as if they are a single bundled signal.

For this purpose, we have defined a composite signal type named XREAL for representing continuous-time analog waveforms as a set of parameters, $\{(a_i,b_i,m_i)'s\}$. As noted earlier, the usage of composite data types to supplement auxiliary information to the signal value is similar to the work in [5], but this work proposes a single composite type for analog signals that can encompass all the different types discussed in [5].

An XREAL signal has three member variables: *param_set*, *t_offset*, and *flag*, as listed in the SystemVerilog type definition in Fig. 3(a). *param_set* is a C-pointer handle that points to a linked list in C that stores the set of parameters. A dynamic data structure such as linked lists was necessary as the number of elements in the s-domain parameter set can vary from signal to signal. Second, *t_offset* is a real-valued variable that indicates the actual time instant of the event. Most logic simulators including SystemVerilog advance the time in finite steps and process events that occur within the same step in an arbitrary order. To circumvent possible causality issues and remove dependency on the fine time step resolution, this additional variable, *t_offset*, is attached to the XREAL signal to indicate exactly when the event has happened. Last, *flag* is an event variable to indicate whether the event change has happened for the signal in the current time step, mostly for the usage within *always* statements. Since the member variable *param_set* that stores the signal value is merely a C-pointer whose address does not change once it is initialized, a separate variable is necessary to notify the blocks consuming this signal that its parameter set has been updated. The block that produces an XREAL signal must trigger this event variable whenever it updates the parameter values (e.g. using the '<-' operator).

An outline of a channel model in SystemVerilog given in Fig. 3(b) may provide some insights on how an analog model is described using this XREAL-type signal. The *always* statement within the module is triggered when the parameter set of the input XREAL signal is updated, indicated by the *in.flag* variable. Then the parameter set of the output XREAL signal is updated according to that of the input signal and the transfer function of the channel, modeled as a linear system. The *eval_system*() function is a DPI function written in C that performs the operation outlined in Eq.(2) as explained in the preceding section. Once the output parameter set is updated, its event variable, *out.flag* is triggered, notifying the subsequent blocks of the change event.

```
a)  typedef struct {
      chandle param_set;
      real t_offset;
      event flag;
    } XREAL;

b)  module channel(
       input xreal in,
       output xreal out);

      chandle TF_chan;  // channel transfer function

      always @(in.flag) begin
          out.param_set=
                  eval_system(in.param_set,TF_chan);
          -> out.flag;
      end
    endmodule
```

Figure 3. a) Implementation of XREAL data type in SystemVerilog and b) module example of channel.

IV. Modeling Decision Feedback Equalizing Receiver

This work uses a decision-feedback equalizer (DFE) as a motivational example to demonstrate the accuracy and speed of the proposed true event-driven simulation methodology. DFE is an equalization technique for high-speed interfaces that subtracts the expected inter-symbol-interference (ISI) incurred by the preceding bits from the current bit signal. Due to the inherent nonlinear characteristics, its analysis usually resorts to time-domain simulation, but its complexity renders the BER analysis with the existing simulators impractical. For instance, as shown in Fig. 4, a DFE receiver consists of linear/nonlinear blocks that consume and produce various signal types including steps, ramps, exponentials, or combinations of all the above. Specifically, a pre-emphasis transmit filter generates a PWL signal, a channel and a linear equalizer give continuous-time smooth waveforms, and a decision feedback filter receives a digital signal from the sampler output and feeds back a series of step signals back to an adder block. The adder combines the waveforms in these different shapes and input the signal to the sampler input for the data decision. In order to evaluate the BER performance of the high-speed link system to a precision level of 10^{-12}~10^{-20}, all of these signals must be accurately expressed and simulated.

The channel and linear equalizer models are both linear filters whose characteristics can be described in frequency-domain transfer functions. As long as the input signals to these models can be expressed in the proposed s-domain form, the output signals can be expressed in the same way and their change events are triggered only once per each input event.

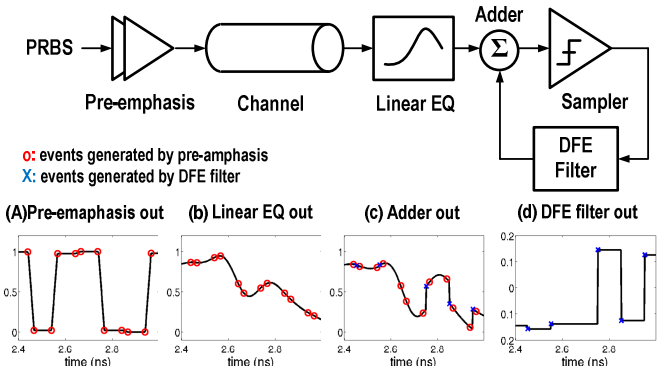

Figure 4. Decision feedback equalizer example. a) pre-emphasis, b) linear equalizer, c) adder, and d) DFE filter output waveforms

Note that this property remains true even for a large number of poles/zeros in the transfer function. Therefore, the simulation costs varies only with the signal's activity.

On the other hand, both the pre-emphasis transmit equalizer and decision feedback filter in Fig. 4 operate as discrete-time FIR filters with the inputs being the digital data. The only difference between two is that the former generates the output with a finite slope (i.e. PWL) while the latter is assumed to generate a signal in ideal steps. As in the linear filter case, only a fixed number of output events (1~2) are generated for each clock trigger event and the proposed basis function can express both types of waveforms without any errors.

The adder combines the events from the linear equalizer and the decision feedback filter. Since the adder is a linear block, its output parameter set is simply a union of the linear equalizer output and the decision feedback filter output. The sampler compares the adder output with a reference voltage at each triggering edge of the clock and generates a digital decision output. It converts a continuous input to a digital output and its output event is updated every time when the clock rises.

V. Experimental Results

The efficiency and accuracy of the proposed method are evaluated with the DFE example described in the previous section. At 10-Gbps operation transmitting 2^7-1 PRBS data pattern, both the pre-emphasis equalizer and decision feedback equalizer has three taps of filter coefficients, the values of each set at {0.974, 0.021, -0.005} and {-0.0284, -0.012, 0.019}, respectively. Frequency-domain transfer function of the channel are estimated with 50 eigen-modes (i.e. poles) from the measured S-parameter model using the *rationalfit* function available from Matlab. The linear equalizer is designed to have one zero at 1 GHz and two poles at 2 GHz and 4 GHz, respectively, to compensate the channel loss which is about 3 dB at 1 GHz.

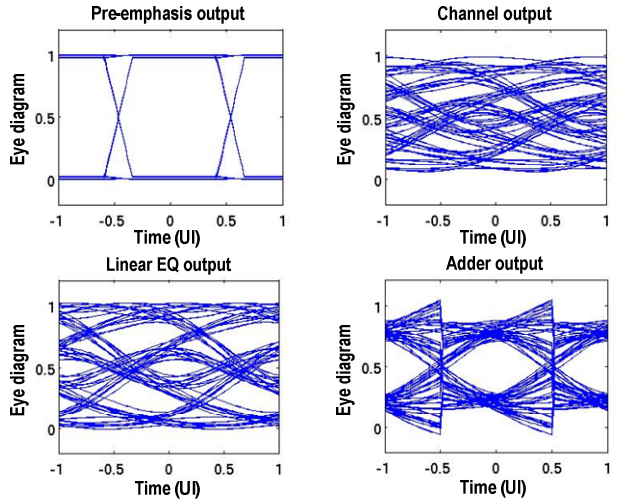

Figure 5. Eye diagrams of a) pre-emphasis output, b) channel output, c) linear equalizer output, and d) adder output

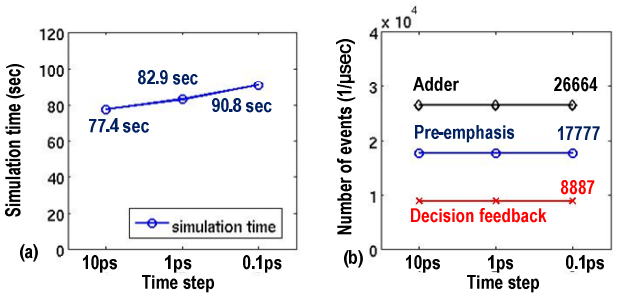

Figure 6. a) Simulation time and b) the number of events in each block vs. simulation time step.

Fig. 5 shows the eye-diagrams of the signals measured at various points of this DFE receiver: the pre-emphasis equalizer, channel, linear equalizer, and adder outputs. These time-domain waveforms can be reconstructed from the parameter sets simply by evaluating the analytical equations.

To demonstrate that the proposed methodology indeed realizes true event-driven simulation, the total simulation time and the number of events generated in each block are measured and shown in Fig. 6. On a linux machine with an AMD Phenom II X4 945 processor, the total times of simulating 100,000 bits were 77.4, 82.9, and 90.8 seconds, when the simulation time steps were 10, 1, and 0.1ps, respectively. Note that simulation time increases very weakly by 17% as the simulation time step becomes finer by a factor of 100. It is because the total number of events processed during the simulation is the same for all cases as shown in Fig. 6(b). Furthermore, both the channel and linear equalizer have the same number of events at their outputs, proving that the output event is generated only when the input event is triggered. In case of the adder block, its output event count is equal to the sum of the two inputs' event counts.

VI. Conclusion

A true event-driven simulation methodology for analog/mixed-signal circuits are proposed and demonstrated on a DFE example. Our choice of the analytical basis function enables the analog/mixed-signal models to generate only a single output event per input event. Also, computing the system responses without numerically integrating the differential equations makes the speed and accuracy of the simulator virtually independent of the time step resolution. As a result, the prototype simulator realized in SystemVerilog can efficiently simulate large-scale systems consisting of both analog and digital components, whose computational costs depend only on the system's activity but not its size.

References

[1] R.A. Cottrell, "Event-driven behavioural simulation of analogue transfer functions," *Proceedings of the European Publication Design Automation Conference*, 1990, pp.240~243.

[2] Y. Wang, C. Van-Meersbergen, H.-W Groh, and S. Heinen, "Event driven analog modeling for the verification of PLL frequency synthesizers," *Behavioral Modeling and Simulation Workshop*, 2009, pp.25~30.

[3] R.B. Staszewski, C. Fernando, and P.T. Balsara, "Event-driven Simulation and modeling of phase noise of an RF oscillator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2005, pp.723~733.

[4] M. Ierssel, H. Yamaguchi, A. Sheikholeslami, H. Tamura, and W.W. Walker, "Event-Driven Modeling of CDR Jitter Induced by Power-Supply Noise, Finite Decision-Circuit Bandwidth, and Channel ISI," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2008, pp.1306~1315.

[5] M.-J. Park, H. Kim, M. Lee, and J. Kim, "Fast and accurate event-driven simulation of mixed-signal systems with data supplementation," *Custom Integrated Circuits Conference (CICC)*, 2011, pp.1~4.