Behavioral modeling of continuous time $\Delta\Sigma$ modulators

Robert Sobot
Simon Fraser University
Email: sobot@cs.sfu.ca

Shawn Stapleton
Simon Fraser University
Email: shawn@cs.sfu.ca

Marek Syrzycki
Simon Fraser University
Email: marek@cs.sfu.ca

Abstract— **A mixed-signal continuous time behavioral model of a continuous time delta-sigma modulator (CT $\Delta\Sigma$) is presented. CT $\Delta\Sigma$ modulators, by their nature, are mixed-signal systems. That fact creates a discontinuity in the traditional IC design flow which assumes that "discrete" and "continuous" time domain designs require separate design tools. In this work, we present a top level behavioral CT $\Delta\Sigma$ model that can be used within the analog IC design environment.**

High speed CT $\Delta\Sigma$ modulators are implemented using both "analog" and "digital" subblocks. We created mixed-signal models of the subblocks in order to efficiently perform simulations that accurately reflect circuit behavior in the continuous time domain. The models were built out of primitives available in $SPICE$ and $Verilog-A^{TM}$.

We present a first order CT low-pass $\Delta\Sigma$ (CTLP $\Delta\Sigma$) as well as a fourth order CT band-pass $\Delta\Sigma$ (CTBP $\Delta\Sigma$) to demonstrate the modeling technique and simulation methodology. We explored the influence of the loop delay and clock jitter on the CT $\Delta\Sigma$ performance.

I. INTRODUCTION

The traditional approach to simulate the behavior of a $\Delta\Sigma$ modulator is by creating z-domain models and using a discrete time (DT) simulator, such as MATLAB^{TM}. This approach is logical since $\Delta\Sigma$ modulators are sampled systems by nature. As a result, most of the $\Delta\Sigma$ implementation were done by using techniques such as switched capacitor and switched current. The increased interest in the high speed continuous time $\Delta\Sigma$ modulators created a problem with the traditional IC design flow, which is not well suited for this application. The principle reason is that the discrete domain simulators were not well connected to the back end IC design tools, while at the same time continuous time simulators were lacking top level behavioral modeling capability. However, mixed-signal simulators are now becoming more practical and are incorporated in the analog IC design environment. Previously published works on $\Delta\Sigma$ modeling were based either on SIMULINK$^{®}$ models [1], custom made C programs, or $Verilog-A^{TM}$ [2].

A description of the basic functional blocks in a CT $\Delta\Sigma$ modulator are presented in Section II. In Section III we present two categories of CT $\Delta\Sigma$ behavioral models: 1) a first order CTLP $\Delta\Sigma$ behavioral model simulation 2) a fourth order CTBP $\Delta\Sigma$ behavioral model simulation.

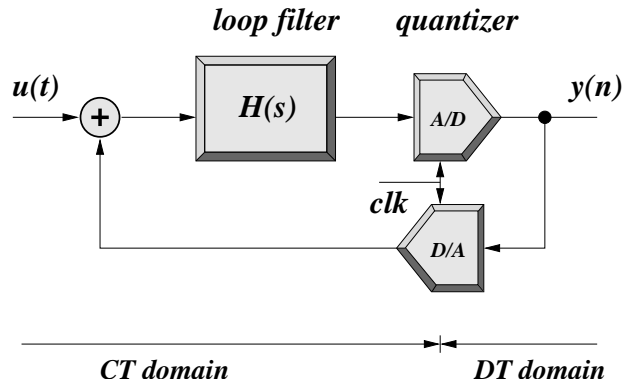Finally, we present concluding remarks on this work in Section IV.

Fig. 1. CT $\Delta\Sigma$ modulator loop architecture

A. General CT $\Delta\Sigma$ modulator loop

A general high speed CT $\Delta\Sigma$ modulator loop architecture is shown in Fig. 1. The loop filter $H(s)$ is typically either low-pass or band-pass. The presence of the high speed clock and quantizer makes the simulation time very long if CT simulators are used, while discrete time simulators use high level models which do not offer enough insight into the circuit behavior.

Today, discrete time IC design tools are well developed making it possible to design very complicated switched capacitor ICs without having to use any of the "continuous time" tools such as $SPICE$. However, in order to design an analog CT IC one must use simulators capable of simulating in the CT domain. A CT $\Delta\Sigma$ modulator is an example of a mixed-signal circuit that creates discontinuities in the IC design flow. It requires a smooth connection between the behavioral models developed in discrete time and the physical realization of the IC circuits. Bridging these two worlds is one of the motivations behind this work.

One method of reducing the simulation time is to create circuit behavioral models using methodologies and tools developed for analog CT IC design. Unfortunately, $SPICE$ based simulators have limited behavioral modeling capability. This problem has been recognized since the early days of analog simulators, which resulted in the development of "mixed-signal" simulators. Fortunately, it is now possible to create behavioral model subblocks using both primitives from $SPICE$ along with the ones from $Verilog-A^{TM}$. This arrangement offers the possibility of reducing the simulation
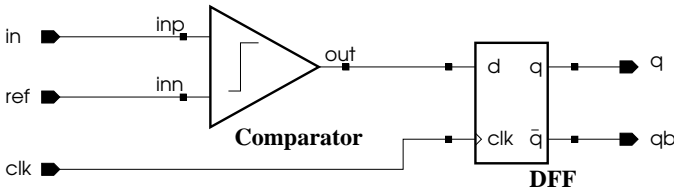
Fig. 2. 1-bit quantizer schematic

Fig. 3. DAC schematic

time by orders of magnitude depending upon the circuit complexity and architecture.

II. BASIC FUNCTIONAL BLOCKS

In this section, we present behavioral models of the basic functional blocks required to create a complete CT $\Delta\Sigma$ modulator. Our models are developed in $SPICE$ and $Verilog-A^{TM}$ [3], while the simulator used was $Spectre^{TM}$ within $Analog-Artist^{TM}$. Our set of basic blocks consist of the following DT/CT behavioral modules: integrator, general s-transfer function, comparator, D flip-flop (DFF), DAC with non-return to zero (NRZ), return to zero (RZ) and hold-return to zero (HZ) pulses, summing block, ideal gain, and clock with controlled jitter. Another important requirement is that all the modules have a controlled propagation delay.

A. Integrator

An integrator is just a k/s function, where k is the integrator gain and $s = j\omega$. It is possible to create a s-domain filter using $SPICE$ behavioral primitives [4], but we prefer to use $Verilog-A^{TM}$ code. Both models have similar simulation times. $Analog-Artist^{TM}$ requires a symbol to be created before generating the veriloga view after that the text editor becomes available for editing.

In our design we used:

```
V(vout) <+  laplace_nd( V(vin),
            [ n0 ], [ d0, d1 ]);
```

to create the k/s function, where n0=k, d0=0, and d1=1. The integrator is sufficient to create loop filters for a CTLP $\Delta\Sigma$ modulator.

B. s-domain transfer function module

A general s-transfer function is needed to create a higher order filter $\Delta\Sigma$ loop function. The, $Verilog-A^{TM}$ laplace_nd function was used to create the transfer function $H(s) = N(s)/D(s)$. Where n0, n1, .. nN are the $N(s)$ polynomial coefficients and d0, d1, .. dD, $(N+1 \leq D)$ for the $D(s)$ polynomial. The mapping methodology between the z-domain and s-domain filter transfer functions and their dependence upon the type of DAC pulse used is not covered here, [5].

C. Comparator

A one bit quantizer can be implemented with a comparator and D flip-flop, Fig. 2. It is possible to create a s-domain comparator using $SPICE$ behavioral primitives [4], but again
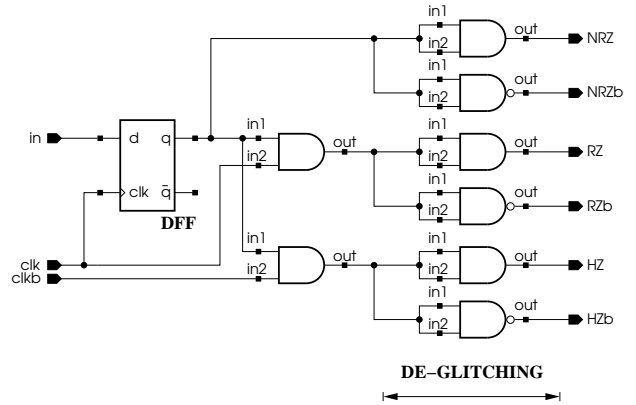
we prefer to use $Verilog-A^{TM}$ code. Both models have similar simulation times. The tanh function was used to simulate maximal signal levels of the comparator. The code that implements the comparator function is:

```
V(out) <+ 0.5*(outHigh-outLow)
          * tanh(slope*(V(in, ref)
          - inOffset))
          + 0.5*(outHigh + outLow);
```

One of the problems with using $SPICE$ to model a comparator is that the *min-max* range used in the $VCVS$ function is not exported by the $Analog-Artist^{TM}$ netlister. The small syntax differences that are not supported by the $Analog-Artist^{TM}$ netlister are important factors that have to be taken into consideration when the goal is to make a versatile model that works both within the $Analog-Artist^{TM}$ and as a stand-alone model.

D. D flip-flop (DFF) and DAC

One of the assumptions in DT analysis is that the pulse is non-return-to-zero (NRZ). However, in CT, the pulse can take on various forms, such as: return-to-zero (RZ), hold-return-to-zero (HZ), and the NRZ, [5], [6], [7]. Also, in implementation there is often a need for inverted pulses $\overline{NRZ}$, $\overline{RZ}$, and $\overline{HZ}$ as well (labeled NRZb, RZb, and HZb on the schematic).

We first created a $Verilog-A^{TM}$ model of the D flip-flop, which is a basic delay unit equivalent to z^{-1}. The key code lines for the D flip-flop:

```
@(cross(V(CLK)-Vth, +1)) x=(V(d) > Vth);
V(q)    <+ transition(voutHigh*x  +
           voutLow*!x, delay, transitTime);
V(qbar) <+ transition(voutHigh*!x +
           voutLow*x,  delay, transitTime);
```

Schematic of the DAC is shown in Fig. 3. It consists of the D flip-flop plus AND digital gates, with the following code:

```
out = (logic1 && logic2) ?
           vlogicHigh : vlogicLow;
V(vout) <+ transition(out, delay,
           trise, tfall);
```
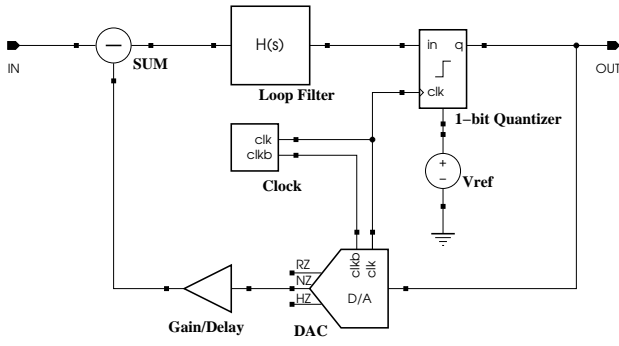
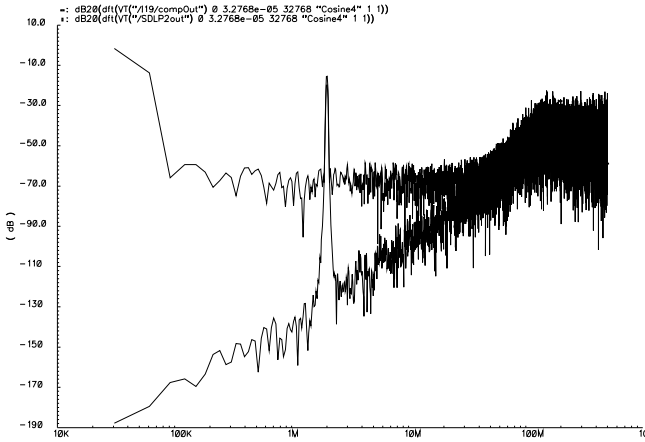Fig. 4. General H(s) $\Delta\Sigma$ modulator schematic

Fig. 5. First order CT LP $\Delta\Sigma$ modulator output

For purposes of our DAC we used `high=1`, and `low=-1`. The D flip-flop delivers NRZ data at the rising edge of the clock. By using both phases of the clock and eight AND/NAND gates we created NRZ, RZ, and HZ data. The total propagation time of the circuit can be controlled at the output gates, while the rest of the gates in the circuit can have a delay time equal to zero.

Finite rising and falling times of the pulse edges cause glitches in switching logic. These glitches are one source of the "rising the noise floor", that can be observed in Fig. 5. The usual practice in CT design is to add "de-glitching" buffers at the interface of DT to CT domains.

E. Summing block

Note that the loop can be designed to work both with the "+" and the "-" sign in the summing block. The summing block with propagation delay code is shown here:

```
V(out) <+ absdelay(V(in1)+V(in2),delay);
```

F. Ideal Gain

Similarly, an ideal gain block with propagation delay code is shown here:

```
V(out) <+ absdelay(V(in)*k,delay);
```

G. Clock modules

The most convenient way to implement an ideal clock that has no jitter is by using the `vpulse` function from $SPICE$. We created a two-phase ideal clock reference by using two `vpulse` sources.

Clock jitter is very important element in CT $\Delta\Sigma$, [7]. Unfortunately, there is no straightforward method to create a clock pulse with added jitter in $SPICE$. However, both MATLAB^{TM} and $Verilog - A^{TM}$ have random number generator function.

The main problem with modeling clock jitter in the time domain is in not knowing the jitter distribution before starting the simulation. Mean value (zero in this case) and variance (one in this case) are defined over the *length of the simulation*, measured in number of the clock cycles. Generating jitter on the clock edges is not doable during the simulation, so some initial work is required. We used `randn` (MATLAB^{TM}) function to generate a vector of random numbers with normal distribution and $var = 1$ over a given number of clock periods.

We choose to generate data for 2^N clock cycles because it simplify the FFT analysis. The $SPICE$ `vpwlf` was used to import this two-column clock data file into the simulations. We generated number of data files with various amounts of jitter incorporated.

III. SIMULATION AND RESULTS

The set of general basic blocks enabled the behavioral model simulation of a CT $\Delta\Sigma$ modulator. The models are generic and used inside $Analog - Artist^{TM}$. The obvious implication of this approach is that in subsequent phases of the IC development we can swap transistor level $SPICE$ models for the various subblocks and evaluate performance with the rest of the circuit being "ideal". Due to the mixed signal nature of the circuits we used $Spectre^{TM}$ for all simulations. In this section, we present simulation results of two categories of the CT $\Delta\Sigma$ modulators: first order low-pass and fourth order band-pass.

A. First order CTLP

We simulated a first order CTLP $\Delta\Sigma$ modulator, Fig. 4, where $H(s) = 1/s$ implements the low-pass filter with a single integrator. Using $1GHz$ clock, $2MHz$ single tone input signal, and running 16384 clock cycles took less then 2min of CPU time on SUN Blade1000 computer to produce FFT plot (using 16384 points and Cosine4 window) shown in Fig. 5. As a comparison, two outputs are plotted: without glitches and jitter, and one with the glitches and jitter.

B. Fourth order CTBP

The same schematic of the general CT $\Delta\Sigma$ loop shown in Fig. 4 was used to simulate the fourth order SDBP $\Delta\Sigma$ modulator where the general s-function block is implemented using a bandpass filter. We used the fourth order bandpass function:

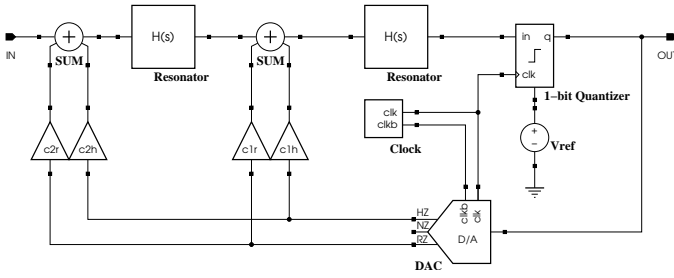$$H(z) = \frac{z^{-2}(2 + z^{-2})}{(1 + z^{-2})^2}$$

Fig. 6. Fourth order CT BP $\Delta\Sigma$ modulator

Fig. 7. Fourth order CT BP $\Delta\Sigma$ modulator output

Fig. 8. SNR vs. clock jitter for 4th order CT BP $\Delta\Sigma$ modulator
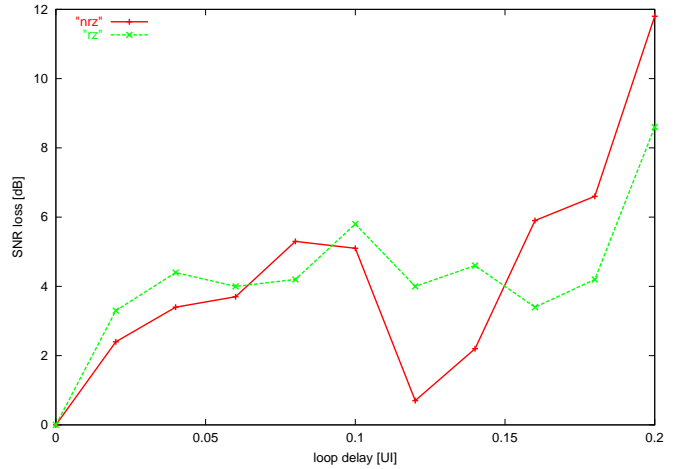
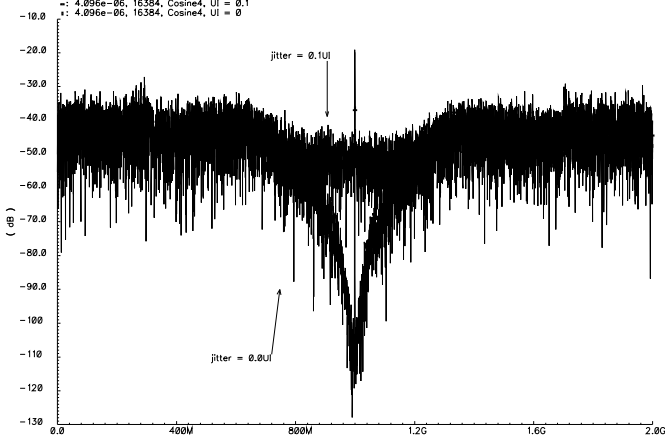Fig. 9. SNR vs. loop delay for 4th order CT BP $\Delta\Sigma$ modulator

which has its counterpart in continuous time (one NRZ clock delay case) [5]:

$$H(s) = \frac{\left(\frac{\pi}{2} - \frac{1}{4}\right)\frac{s^3}{T} + \left(\frac{3\pi^2}{16} + \frac{\pi}{4}\right)\frac{s^2}{T^2} + \left(\frac{\pi^3}{8} + \frac{\pi^2}{16}\right)\frac{s}{T^3} + \frac{3}{4}\left(\frac{\pi}{2T}\right)^4}{\left(s^2 + \left(\frac{\pi}{2T}\right)^2\right)^2}$$

This BP implementation is an $f_s/4$ structure, we used $1\,GHz$ signal sampled by a $4\,GHz$ clock over $4096\,ns$. The $\Delta\Sigma$ modulator was simulated with and without jitter on the clock, the output plot is shown in Fig. 7 for jitter of $UI = 0$ and $UI = 0.1$. One elegant way of actual implementation of the general fourth order bandpass function in s-domain is shown in Fig. 6 [6] [5].

This structure can be implemented with one RZ clock delay as well. Plot showing the maximal SNR vs clock jitter for the two cases is shown in Fig. 8, while the SNR vs. loop delay is shown in Fig. 9.

IV. CONCLUDING REMARKS

A mixed-signal behavioral model for a CT $\Delta\Sigma$ modulator circuit has been presented. It has been shown that by using mixed-signal approach for behavioral modeling one can achieve high simulation speed and produce meaningful results by staying within one design environment throughout the design process. It should be noted that simulations can be accelerated by shifting delays to clocked blocks. A mixture of $Ver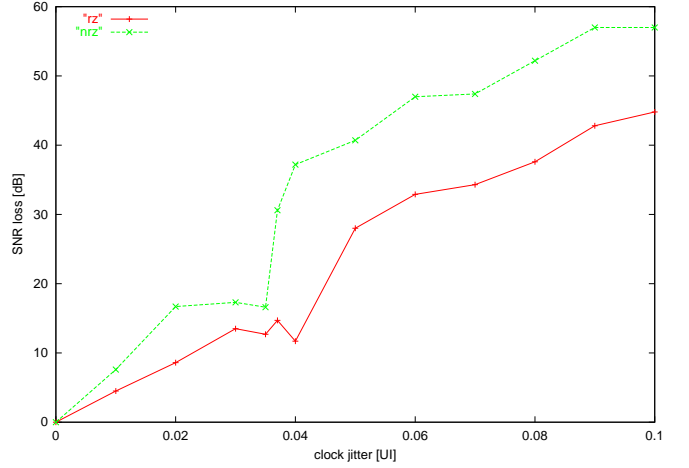ilog-A^{TM}$ and $SPICE$ models makes possible rapid behavioral level simulations within the $Analog-Artist^{TM}$ environment used by analog IC designers.

REFERENCES

[1] S. Brigati *et al.*, "Modeling sigma-delta modulator non-idealities in SIMULINK®," in *Proceedings of the IEEE International Symposium on Circuits and Systems 1999, ISCAS '99*, May 1999, pp. 384–387.

[2] M. Vogels, B. D. Smedt, and G. Gielen, "Modeling and simulation of a Sigma-Delta digital to analog converter using VHDL-AMS," in *Proceedings 2000 IEEE/ACM International Workshop on Behavioral Modeling and Simulation, BMAS 2000*, 2000, pp. 5–9.

[3] D. Fitzpatrick and I. Miller, *Analog Behavioral Modeling with the Verilog-A Language*. Kluwer Academic Publishers, 1998, no. 0-7923-8044-4.

[4] J. A. Connelly and P. Choi, *Macromodeling with SPICE*. Prentice Hall, 1992, no. 0-13-544941-3.

[5] O. Shoaei and W. M. Snelgrove, "Design and Implementation of a Tunable 40MHz-70MHz Gm-C Bandpass $\Delta\Sigma$ Modulator," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 7, pp. 521–530, July 1997.

[6] O. Shoaei, "Continuous-time delta-sigma a/d converters for high speed applications," Ph.D. dissertation, Carleton University, Ottawa, Canada, 1995.

[7] J. A. Cherry and W. M. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*. Kluwer Academic Publishers, 2000, no. 0-7923-8625-6.