High-level design case of a switched-capacitor low-pass filter using Verilog-A

Erik Lauwers, Koen Lampaert*, Paolo Miliozzi*, Georges Gielen

Katholieke Universiteit Leuven
ESAT-MICAS
Leuven, Belgium
erik.lauwers@esat.kuleuven.ac.be

** Conexant Systems Inc.*
Newport Beach, CA, U.S.A.

Abstract

System design requires experienced designers that use heuristics and built up knowledge to propose a high order solution. Behavioral models can help to formalise, optimise and speed up this design cycle. A design case is presented that shows how behavioral models are used to support system design. Models of two basic analog functions (operational amplifier and switch) are developed in Verilog-A and used in the design of a 1st-order switched-capacitor low-pass filter. This allows to find a first-order system solution on a higher level than the fully designed transistor schematic. The specifications of the subblocks can then rapidly be refined into a transistor netlist by an analog designer.

1. Introduction

Reuse of analog microelectronic building blocks is common practice. A designer picks a design from a previous project, performs some minor changes to suit his actual specifications and moves on to the next design task. Clearly a well known and fast design cycle. However, if the actual specifications differ too much from the old ones or if the technology changes are too important, the design time becomes comparable to the case where the new design is made from scratch.

What happens when not small building blocks like opamps but small analog systems have to be built such as for example a switched-capacitor filter (Figure 1)? In this case, the nominal design parameters for the operational amplifier (opamp) need to be changed for each individual design iteration. If the effect of a different gain-bandwidth product, for example, on the filter output has to be checked, a new nominal opamp design has to be created making iterations such as this very time consuming.

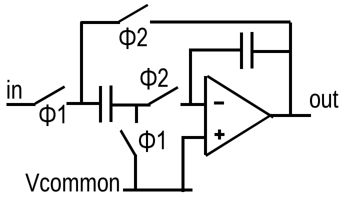

Figure 1 Switched-capacitor filter example

For this reason, it is good to have a high-level behavioral model available of an opamp to find the specifications that lead to a good overall filter performance. When first-order nominal specifications are obtained in this way, then a sized transistor netlist of the opamp can be generated that is most likely close to the final solution.

This paper describes the Verilog-A solution that was developed to tackle the above mentioned design case problem. Also, experiencesare reported of possible problems related to the use of a behavioral language and more specifically Verilog-A for mixed behavioral/schematic simulations.

In section 2 an overview of the design case is given. Section 3 covers the Verilog-A models of the opamp and the switches. In section 4 simulation results are presented. Some more typical simulation and modelling issues of

Verilog-A are covered in section 5 and finally the conclusions are presented in section 6.

2. A first-order switched-capacitor low-pass filter

In Figure 1 a single-ended version of the designed switched-capacitor (SC) filter is drawn. The real filter (Figure 2) is a differential implementation of this topology. A different symbol is used for the one-transistor switches and the passgates.

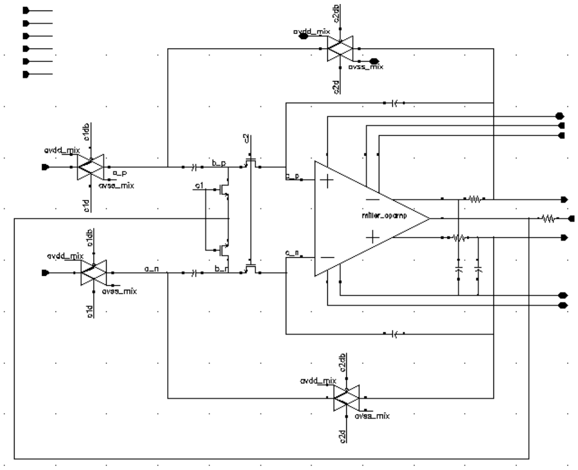

Figure 2 Schematic of fully differential 1^{st}-order SC LPF

The design of a filter like this can be split into different stages. The first stage, the most creative one, covers the topology selection and selection of the capacitor sizes. In this stage ideal switches and an ideal opamp are used. Next, the switches and the opamp have to be sized to meet the required performance. Based on experience and hand formulas high-level specifications of the opamp are set as goal for the transistor design. If at this point it is still not clear whether the filter structure will work properly, this design approach easily becomes time consuming. To avoid iterations of the detailed transistor design before the high-level filter design has been completely satisfactorily, a behavioral model for the opamp is used. This helps to find a first-order estimate of the specifications and allows fast trade-off studies between different filter topologies. Also, for the switches a behavioral model is used. This allows a study of the impact of separate parasitic effects of the switches on the filter performance while the other parts of the filter can be nearly ideal (a behavioral model) or real components.

For the case study reported in this paper, a Miller opamp is modelled. This does not form a hard limitation on any specification if only the correct behavior is wanted. If additionally some kind of circuit synthesis is desired in the form of 1^{st}-order design parameters such as currents and transconductance values, this forms a limitation. In this case, more opamps have to be modelled. The behavioral models of the design case are described in the next section.

3. Behavioral models

Two models are used: one model for the switches and one model for a Miller opamp.

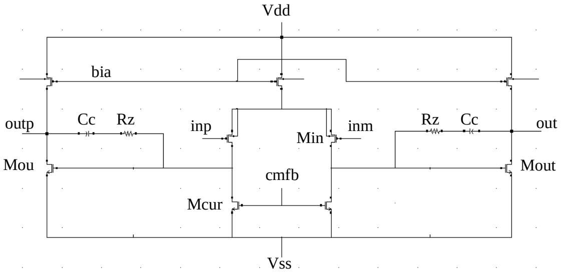

Figure 3 Schematic of a Miller opamp

Miller opamp model

The schematic of a two-stage Miller-compensated opamp is well documented in the literature (e.g. [1]) and is repeated in Figure 3. First a selection of desired high-level parameters and model capabilities has to be made. This is always a trade-off between model detail and exact SPICE-like behavior on the one hand, and generality and model-development time on the other hand. As a general rule it is good not to forget that the aim of the model is to decrease the design cycle time and not to make the actual design. First-order system simulation is the goal and thus the simpler the model the better.

The high-level parameters selected are: gain-bandwidth (GBW), gain, slew-rate (SR), total integrated output white noise, phase margin

(PM), input referred offset (Vos), capacitive load (CL) and resistive load (RL). Extra capability of the model is to generate noise at the in/outputs during a noise analysis. The user defines the white noise power level as one of the input specifications and can optionally specify a corner frequency.

To obtain the correct AC behavior the core of the behavioral model is the well known small-signal model of the Miller opamp as shown in Figure 4.

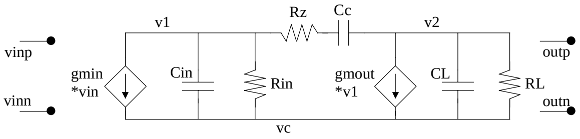

Figure 4 Small-signal model of Miller opamp

The component values are calculated from the high-level parameters during the “initial block” of the Verilog-A model. This implies a certain rough sizing of the opamp which is achieved by following some kind of design plan, that is specific to the topology used here.

For the noise analysis the same AC model is used. During a transient simulation, however, many effects come into consideration that are unimportant for the AC simulation. These effects are slewing and clipping effects combined with nonlinear transistor effects. The differences are programmed in specific “AC” and “TRAN” conditional simulation blocks.

Slewing is taken into account by simply limiting the current in each stage. For example, slewing in the input differential pair is limited by the tail current. Combined with the typical tanh form of the differential output current a full non-ideal 1st-order current behavior is put into the model. The current through the output amplifying stage also has to be modelled in more detail. The constant current-source current is combined with the quadratic voltage-current behavior of the amplifying transistor Mout. Depending on the voltage of node v1, these currents are then combined to obtain the total differential output current. Expressed in a formula, the output AC-current is a combination of the DC-current Iout and:

$$\mathrm{Iout}\cdot\left[\frac{\left(\mathrm{v1}/2\right)}{\left(\mathrm{V_{GS}}-\mathrm{V_T}\right)_{\mathrm{out}}}\right]^2+\frac{2\cdot\left(\mathrm{v1}/2\right)\cdot\mathrm{Iout}}{\left(\mathrm{V_{GS}}-\mathrm{V_T}\right)_{\mathrm{out}}}\qquad(1)$$

Switching between different combinations occurs when (v1/2) crosses $(V_{GS}-V_T)_{out}$ or when one of the amplifying transistors is switched off.

Clipping has to occur on both (internal) nodes and these are separate effects. The clipping on node v2, which is not equal to the output nodes (see below), is the most obvious effect since the outputs can not be smaller or larger than the supply voltages. Clipping has to be considered separately for each output because this depends on the output common-mode voltage. This clipping can be made more accurate by including the effect of the output transistors going into their linear region of operation. The output clipping is realized by introducing a voltage-driven voltage source or by copying the v2 voltage. This is necessary to avoid local feedback and to avoid interaction between the circuitry connecting to the model and the wanted model behavior. This will be explained in more detail later.

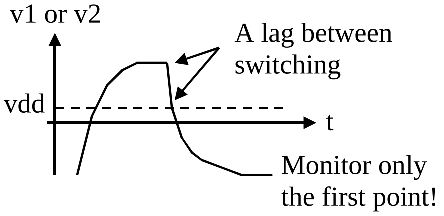

Figure 5 Large-signal clipping behavior

The clipping on the internal node v1 is less obvious, but is needed with respect to transient behavior for large input signals or for switching behavior (e.g. digital input signals). Figure 5 shows that for a large input signal, the voltage on node v1 can become very large, depending on many factors such as for example the gain required by the user. The output at node v2 in this example is clipped at Vdd. When the input voltage is switched, it takes some time before v2 dives under Vdd and hence before the output starts changing. This delay is not correct and must be eliminated. When the absolute voltage at node v2 becomes larger than twice the supply voltage, a counter current is injected into node v1. Note that sensing the voltage at node v1 and injecting a counter current is not a good solution. This will lead to an unpredictable voltage level at node v1. This is true in general. Every time feedback is used, one has to check for correct voltage levels and take into account that simulation time will increase, sometimes even tremendously.

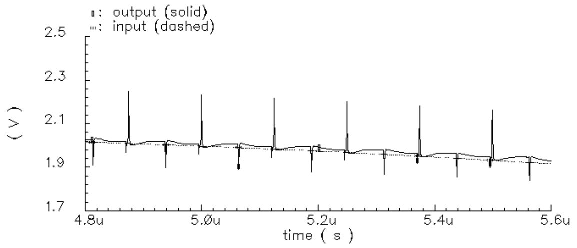

Figure 6 Transient simulation of the 1st-order SC LP filter (full transistor schematic)

Finally the interaction with the surrounding circuitry has to be modelled. Specifically for the case of a SC filter where glitches coming from the switches are present (Figure 6). It is important to keep in mind that any element present in the behavioral model is translated by the simulator into the total system matrix. For a correct behavior, independent of the surrounding circuitry and with a correct user provided load, this means that some decoupling with a voltage-controlled source has to be foreseen. A controlled voltage source overrules voltage glitches at the output, hence a controlled current source with a parallel resistor is used. If the parallel resistor equals RL, the output impedance of the opamp as seen by the surrounding circuitry is correct.

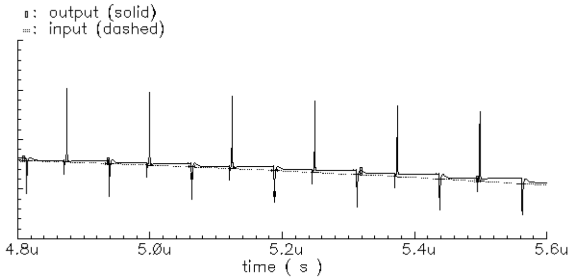

Figure 7 Transient simulation of the SC LPF with a behavioral opamp

To make sure that the typical common-mode glitches are attenuated by the common-mode feedback loop present in the full opamp circuit, the common-mode output voltage must be fed back to node v2. In Figure 7 the result of a transient simulation of the total filter using the behavioral opamp model is shown. The result resembles the full transistor schematic simulation result (Figure 6) very well.

Switch model

The switch is modelled with three extra non-ideal effects compared to an ideal model. The ideal model, that can readily be found in literature (e.g. [2]), sets the voltage across the switch to zero when it is closed. When the switch is opened, the current through the switch is set to zero. The three effects included in our model are:

- control-signal feedthrough
- signal-dependent opening/closing
- a channel resistance

Modelling a channel resistance introduces an important difference compared to the ideal model. Now a continuously varying channel resistance value, within one state of the switch, is abruptly changed to another value when the state changes. This is a discontinuity and has to be taken care of in the model in order to ensure convergence during simulation. The most straightforward solution is to use the “transition” operator of Verilog-A. However, due to the specific implementation of this operator, the simulation is slowed down enormously. A possible solution implemented in the model is to send the resistance through a low-pass RC filter and to use the output as the effective channel resistance. Although still slowing down the simulation, the time loss is smaller.

Some other remarks concern the DC convergence of the switch model. For the simulator to find a DC solution in all situations, including for example a purely capacitive output, the output of the switch has to be given an initial value. A good initial value is to equal the output voltage to the input voltage or to state in the model code that the voltage difference between input and output is zero:

$$\mathrm{V(out,in)} < + \, 0.0; \qquad (2)$$

This however introduces in the system matrix a short connection between the input and output. The solution is to "probe" the input voltage and to set the output voltage equal to it:

$$V(out) < + V(in); \qquad (3)$$

Using this solution for DC convergence means that the direction in which the switch is inserted into the circuit matters. This solution only works when the input is a known DC voltage. Since always one side of the switch in our SC circuit fulfills this requirement, flipping the switch solves the problem without altering the transient behavior.

4. Switched-capacitor simulations

The opamp and switch behavioral models are now used in the selected SC filter topology. Typical specifications are a minimal spurious free dynamic range (SFDR) and a maximal clock tone at the clock frequency for a given input signal. In Table 1 an overview of the simulated values is given.

Table 1 overview of filter simulation results

Simulation results	SFDR [dB]	Clock tone [dB]	Simulation time
Full schematic	75.9	42	100%
Beh.opamp	71.7	41.2	84%
Beh.passg. and opamp	72.3	42.1	164%
Beh.passg.	80.5	41.4	154%
Full behav.	71	41.8	169%

A distinction is made between passgate switches that transmit the signal and other switches that are always resetting voltages to the same reference voltage (see Figure 2). The resulting values using behavioral models are close to the full schematic results (within 10%). This is acceptable for a 1st-order high-level design approach. Making a detailed opamp design using the obtained high-level parameters is the next step from here. Note also that emphasis must go to a gain in overall design time and not in simulation time. As a matter of fact, the behavioral models simulate up to 70% slower.

To check the sensitivity of the approach, a behavioral opamp was used with a GBW 5 times smaller than the GBW used in the simulation examples of Table 1. The result was a decrease (as expected) in SFDR to 61dB.

5. High-level design using Verilog-A

Some general remarks related to the use of Verilog-A as a high-level design aid are provided here. They were encountered during the SC design case but can be generalized. For making a high-level model, 2 approaches are possible. The first is making a hierarchical behavioral model using the different models as subblocks. However, in CADENCE the parameters of the subblocks have to be available at compile time and cannot be set any more during runtime in for example the initial block. In the Verilog-A language reference manual a command (defparams) to handle this problem is defined but it is at this time not yet implemented in SPECTRE. The second approach is putting into a schematic all the different behavioral subblocks. However, when working with separate modules, it is common to obtain a rigid loop of voltage sources for the DC solution and hence convergence problems. Certainly with switches present, this is a regular problem. This can easily be tackled by introducing 1mΩ resistors in the branches where the problem occurs.

When comparing a hardware description language to for example a scripting language as Matlab^{TM} for use as a high-level design environment, it is the author's belief that a HDL offers many more opportunities. The biggest one being of course the possibility to mix transistors with idealized modules. However, the problem is that debugging is very cumbersome. Cryptic error messages such as "arithmetic exception" or completely wrong results are common (at least in the current Verilog-A implementation) and ask a skilled language knowledge that most designers don't have the dedication and time to develop.

6. Conclusions

Behavioral models of an operational amplifier and a switch have been developed and implemented in Verilog-A, and used to find a first-order estimation of the subblock specifications of a switched-capacitor filter. The differences in the simulation results using the behavioral models or the fully designed filter are never larger than 10%. After this first design cycle, the specifications of the opamp can be refined into a fully sized transistor netlist. The net gain is a reduced overall design time, certainly when different topology solution trade-offs are included in the design.

Throughout the paper attention is given to encountered problems related to the use of Verilog-A for modeling and mixed behavioral/ schematic simulations.

References

[1] K.Laker, W.Sansen, “Design of analog integrated circuits and systems”, McGraw-Hill, ISBN 0-07-113458-1

[2] http://www.eda.org/verilog-ams/htmlpages/sample_lib.bedm.html#Ideal_Switch

[3] http://www.ovi.org