

Selection of the simulation approach for  
 $\Delta\Sigma$  modulators

*A SYSCONV IC&D deliverable*



**K.U. Leuven**  
**ESAT – MICAS**

M. Vogels  
G. Gielen

# Contents

1	Introduction . . . . .	2
2	Different simulation types . . . . .	3
2.1	Small signal analysis . . . . .	3
2.2	Sampled data simulation . . . . .	4
2.3	Macromodel simulation . . . . .	5
2.4	Full circuit simulation . . . . .	5
3	Implementation of the simulation method . . . . .	7
4	Conclusions . . . . .	8

## **1 Introduction**

In present day design of complete systems on a single chip, there is an increasing demand for systematic, top-down design. This enables the designer to choose the right parameters for the different building blocks of the system, starting from the system specifications. In this way, systematic design will increase the chance of a first-time-right design and therefore a faster time-to-market. However, to make systematic design possible, accurate high-levels models and fast simulation methods are necessary to gain insight in the behavior of the system and to provide specifications of the system's building blocks. Since complete systems on a chip will normally be mixed signal systems, fast and high-resolution analog to digital conversion remains an important research topic. An interesting choice for the implementation of an analog to digital converter is a Delta-Sigma modulator, since this type of circuit provides a robust and inexpensive way to perform accurate analog to digital conversion [1]. However, the oversampled nature of the Delta-Sigma modulator is problematic for high-level simulations since the number of output-samples needed to gain accurate insight in the system's behavior is large due to the oversampling ratio. This means that fast simulation of Delta-Sigma modulators will always be at the expense of a lower complexity of the macro-model. This deliverable addresses this problem by first comparing a number of different simulation methods in section 2. However not only the simulation method is of importance, also the way it is implemented proves to be a factor that has to be taken into account. This is done in section 3. Finally some conclusions have been drawn in section 4.

## 2 Different simulation types

A number of different simulation methods can be used to simulate Delta-Sigma modulators. However, simulation is complicated due to the large number of output samples which is needed to resolve the signal to noise ratio. Furthermore, the presence of a strong non-linearity (the quantizer) forces us to use other methods than standard small signal analysis employing linearized models. A number of different simulation methods can be distinguished.

### 2.1 Small signal analysis

As stated before, the strong nonlinearity makes it hard to use this type of simulation method. However, by replacing the quantizer by an amplifier with a fixed gain  $K$  (with an additional phase shift [2]) and a uniform noise source, such as is done in Fig. 1, some of the nonidealities influencing the behavior of the Delta-Sigma modulator can be examined. The effect of nonideal integration and nonideal switching can be investigated to some extent, although typical effects such as overloading and input signal related stability cannot be simulated.

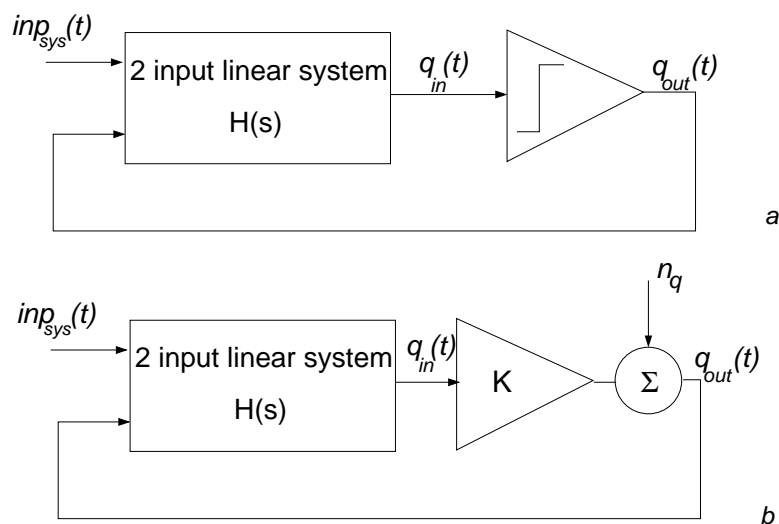


Figure 1: Basic Delta-Sigma modulator (a) and its linear model (b)

## 2.2 Sampled data simulation

A very popular technique for the simulation of switched capacitor circuits (eg. Fig 2) is sampled data simulation [3]. Here one makes use of the fact that all events in the circuits

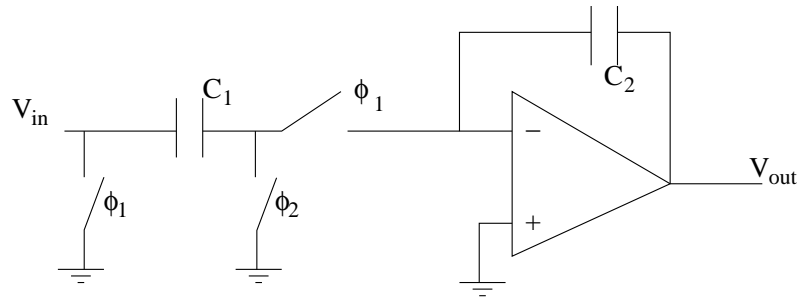


Figure 2: Basic switched capacitor integrator

are triggered by a clock signal. This means that although charge is transferred during one half of the clock cycle, the result of this charge transfer will only be noticed at the rising (or falling) clock edge. This implies that a description can be found as if charge was only transferred at the rising (or falling) clock edge. Fig 3 shows an example of this. As clock  $\phi_2$  is high, the corresponding switch is closed. Capacitor  $C_1$  becomes charged, while the charge on capacitor  $C_2$  is not changed. If clock  $\phi_1$  becomes high (assuming non overlapping clocks), the charge of capacitor  $C_1$  is stored on capacitor  $C_2$ . This means that at the end of the two clock cycles (i.e. the falling edge of  $\phi_1$ ) the charge on  $C_2$  is changed by an amount that can be calculated on beforehand and no knowledge is needed on the charge at intermediate time points. Nonidealities can be incorporated by investigating how

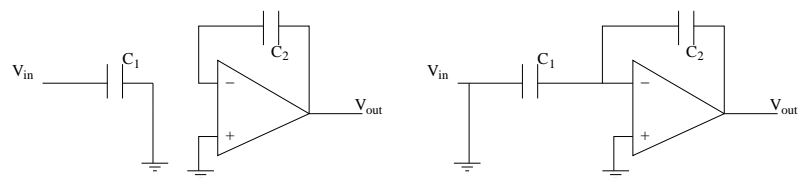


Figure 3: Sampling and integration phase of the switched capacitor circuit

the charge of the integration capacitor (at the clock event), is modified by the nonidealities. Examples of this procedure can be found in [4, 5]. The resulting nonideal sampled data models allow the use of discrete-time simulation which greatly improves the simulation time.

### 2.3 Macromodel simulation

Macromodel simulation is a widely used method of improving simulation time of analog blocks. Fig 4 shows a macromodel of an OTA, for which some linear nonidealities are present. This method uses the fact that a large number of nonidealities present in the system can be modeled in a relative simple way, while the accuracy is not compromised. Simulation is usually done using a SPICE-like simulator capable of handling macromodels. Although the simulation is slower than the methods described above, it has the benefit that the modeling of most nonidealities is trivial, which means that the (re)modeling phase takes much less time.

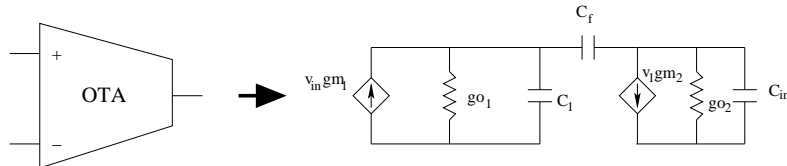


Figure 4: Basic Delta-Sigma modulator (a) and its linear model (b)

### 2.4 Full circuit simulation

The final simulation method that will be described here is full circuit simulation. Here no modeling is done, the circuit is just simulated using transistor models. Simulations like this will take a very long time and are usually only performed at the end of the design cycle, to get insight on the performance of the circuit with all parasitics and nonidealities present.

To be able to select the most suitable simulation method for Delta-Sigma modulators a

Table 1: Comparison of different simulation methods.

<b>Method:</b>	<b>Linear</b>	<b>Sampled data</b>	<b>Macromodel</b>	<b>Full Circuit</b>
<b>Simulation speed</b>	very fast	fast	slow	very slow
<b>Simulation accuracy</b>	low	good	good	very good
<b>Modeling effort</b>	reasonable	high	low	negligible

summary of the most important aspects of the different methods mention above is stated in table 1.

### 3 Implementation of the simulation method

From table 1 it is clear that the sampled data simulation method combines the benefit of fast simulation with a reasonable high accuracy. Although the modeling task is quite large, still this method is preferable over the other methods. For the implementation of this method a number of options are available. First of all there is the widely used Matlab, a general purpose mathematical tool. Although sampled data simulation can be implemented relative easily in Matlab, the execution speed is very low, since Matlab code is interpreted rather than compiled.

A second choice are the standardized languages VHDL or VHDL-AMS. The advantage is that the code will be compiled and therefore simulation should be much faster than using Matlab. The use of a standardized language makes it possible to make the  $\Delta\Sigma$  implementation part of a complete system simulation and makes the exchangeability of the code much easier. The implementation in VHDL-AMS will likely take a bit longer, due to the more complex nature of the code.

A third alternative is of course the implementation in a programming language (such as C). Here the advantage is that the programmer is in complete control and the implementation becomes very efficient. This is very important, if the simulator is put inside a optimization loop or if sweeps with multiple variables need to be done.

A fourth way of simulation  $\Delta\Sigma$  modulators was developed during the Sysconv project. By using the so-called RoCFaST algorithm [6] an additional speed-up of a factor four is possible over a dedicated C implementation. The problem here is that the modeling and implementation time are very large, making hard to implement additional nonidealities and topologies

Figure 5, gives a comparison of the speed of the different implementation methods. Here all speeds are normalized with respect to the C-implementation of Daisy [7].

Table 2 gives an overview of the different implementation methods mentioned above.

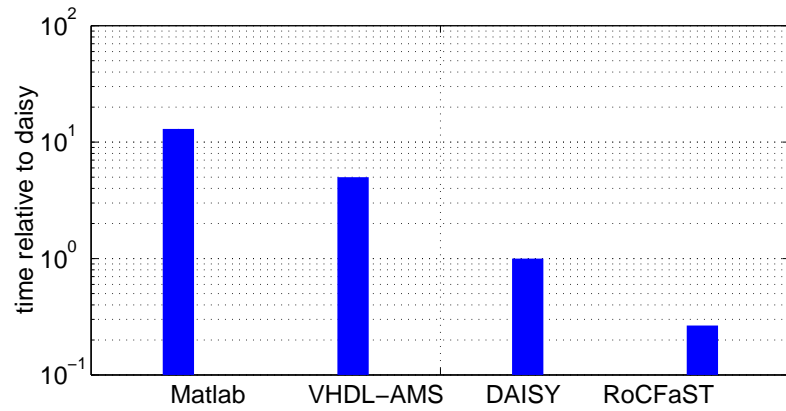


Figure 5: Speed comparison of different implementations

Table 2: Comparison of different implementations of the sampled data method.

Implementation:	Matlab	VHDL(-AMS)	C-code	RoCFaST
Simulation speed	slow	reasonable	fast	very fast
Implementation effort	low	reasonable	high	very high

## 4 Conclusions

In this deliverable the different methods available for simulation of  $\Delta\Sigma$  modulators were described. A comparison between these methods shows that the sampled-data algorithm is the most promising, especially for use in optimization loops. For the implementation of this algorithm, there are also a number of possibilities. Here the trade-off is between the effort it costs to implement the algorithm in the specific code and the execution speed. A very good choice is a C implementation. Although some time is lost implementing it, a large increase in simulation speed is obtained. If there is need for even more speed, the RoCFaST algorithm can be used, although there the implementation effort will sometimes be higher than the speed gained.

# Bibliography

- [1] J. Candy and G. C. Temes, “Oversampling methods for A/D and D/A conversion,” in *Oversampling  $\Delta\Sigma$  converters*, pp. 1–25, IEEE Press, 1992.
- [2] J. van Engelen and R. van der Plassche, “New stability criteria for the design of low-pass sigma-delta modulators,” in *Proceedings of the ISLPED*, pp. 114–118, August 1997. USA.
- [3] A. Opal, “Sampled Data Simulation of Linear and Nonlinear Circuits,” *IEEE Trans on CAD*, vol. 15, pp. 295–306, 1996.
- [4] K. Francken and G. Gielen, “SYSCONV deliverable 1.1.1: Analysis of major nonidealities of multibit oversampling converters,” tech. rep., Katholieke Universiteit Leuven, 1999.
- [5] K. Francken and G. Gielen, “Optimum system-level design of delta-sigma modulators,” in *ProRISC*, 1999.
- [6] M. Vogels and G. Gielen, “SYSCONV deliverable 1.2.2a: Models for oversampling converters,” tech. rep., Katholieke Universiteit Leuven, 2001.
- [7] K. Francken, P. Vancorenland, and G. Gielen, “Daisy: A simulation-based high-level synthesis tool for delta-sigma modulators,” *Proceedings International Conference on Computer Aided Design*, pp. 188–192, 2000.