VIRTUAL TEST BENCH FOR DESIGN AND SIMULATION OF DATA CONVERTERS

P. Estrada[1], F. Maloberti[2]

1. Texas A&M University, College Station, Texas, USA.
2. University of Pavia, Pavia, Italy and University of Texas at Dallas, USA.

Abstract

This work presents a simulation environment for the design of data converters using behavioral modeling and the MATLAB- SIMULINK platform. This environment utilizes a graphic user interface as a CAD tool to design and simulate different data converter architectures. Post-processing analysis tools are included for static and dynamic performance calculation.

Introduction

A data converter system contains many different parts, both digital and analog. For high resolution and for high conversion speed the complexity of systems becomes very high. In order to handle the global design it is therefore necessary to use a top-down approach. Starting from the specifications the architecture and the basic cells are defined and designed. The top-down approach must be followed by a bottom-up verification. If the required specifications are not met, there is an adjustment of individual block specifications, partitioning and architecture. The use of behavioral modeling significantly contributes to the design process: an accurate description of data converter behavior permits us to reduce the computational efforts translating into shorter simulation time.

This paper describes a behavioral design environment addressing and supporting the following design steps:

- Selection of the architecture.
- Verification that the architecture is suitable.
- Estimation of the limits coming from each block.

In summary, the developed user-friendly environment allows the user to design a data converter and define its basic block specifications at system level. By using behavioral models, it significantly reduces the simulation time and, for educational purposes, with behavioral description it helps in better understanding architectures and the limitations associated to the real blocks used.

Simulation Environment Considerations

Behavioral modeling is a problem that can be approached differently depending on the type of system that we want to simulate. In the case of mixed-signal systems, there are several languages or tools available that allow us to construct behavioral models. For instance, MATLAB has the SIMULINK toolbox in order to describe continuous and discrete systems in a graphical environment. MATLAB can perform analog and digital simulations. The difference between them is that in the analog simulation, the tool solves systems whose values can change continuously during simulation, while digital simulation is event-driven, and system values can change only at prescribed times. This language can deal with both types of simulation in the same system, making it ideal for mixed-signal behavioral modeling. The language has also provided the ability to develop in an easy way Graphic User Interfaces (GUI) to interact with the user in a graphical way. By using GUIs, the simulation environment becomes very intuitive and easy to use, as opposed to having to learn a set of commands and functions developed by the tool.

Data converter simulation requires aside of the model itself, suitable input signal generators as well as post-processing tools for calculating the performance of the converter. The typical test signal used for characterizing data converters is a sinusoid or a ramp, which are easy to implement in any computer simulation environment. Also in many cases white noise sources are required. In fact, to properly model data converters, the thermal noise produced by resistors, switches and operational amplifiers has to be considered [1]. Since a data converter is a sampled data system, the aliasing effect of the thermal noise has to be taken into account.

Once we have a model of the data converter and the proper input signals, a simulation of the circuit in the time domain provides a sequence of sampled data values. Then, a suitable processing of the raw output data permits us to evaluate the performance of the data converter. Typically, depending on the data converter architecture, we are interested in the linearity parameters (integral non-linearity or *INL* and differential non-linearity or *DNL*) or in the resolution parameters (effective number of bits *or* N_{eff}, signal-to-noise ratio or *SNR* and the signal-to-noise and distortion ratio or *SND*R).

Basic Blocks

The most common data converter architectures used are the

successive approximation algorithm, the flash approach, the sigma delta technique and all the algorithms that can be achieved with pipeline implementations. The basic blocks that are found in these architectures can also be found in the architectures for other conversion algorithms.

The successive approximation converter commonly uses the charge redistribution architecture. This architecture has two basic blocks, a capacitive array controlled by switches and a comparator. The flash converter utilizes a passive resistive network to generate reference voltages and comparators. The sigma delta uses switched capacitor integrators, a comparator and complex digital circuitry. DAC and pipeline architectures use operational amplifiers (to achieve amplification by a given factor or to subtract voltages) and, again comparators. Therefore we will discuss the operational amplifier, and, in particular, its use in switched capacitor integrators, the comparator and the resistive array used in flash converters.

Operational Amplifier

Operational amplifiers are key components in data converter circuits. Quite often the performance of the operational amplifiers bounds the performance of a complete data converter. It is therefore necessary to include an accurate model of the operational amplifier, considering all of the non-idealities. Linear parameters such as finite gain and bandwidth are considered, as well as the non-linear effects, such as (hard) saturation and slew-rate.

Since data converters are sampled-data systems, there are two possible approaches for modeling operational amplifiers. The first approach is based on traditional models of the operational amplifiers. The models consist of a set of equations and differential equations, which describe the behavior of the circuit. In the simulation, the transient behavior of the circuit is considered for each clock cycle. The simulation obviously allows us to obtain a good accuracy, but at the expense of a long simulation time. The second approach is based on models of the complete sub-circuit (for example an integrator or a buffer), which includes the operational amplifier. The model doesn't perform the time simulation each clock cycle but uses given equations that account for the global effect of the operational amplifier non-idealities at the end of each clock cycle. Therefore, the use of relatively simple behavioral equations permits us to estimate the error produced at the output of the sub-circuit without going into the details of the transient behavior within the clock cycle. This approach is of course less accurate than the previous one, but much faster [2].

As an example, we can consider a switched capacitor (SC) integrator with transfer function

$$H(z)=\frac{z^{-1}}{1-z^{-1}}. \qquad (1)$$

Analog circuit implementations of the integrator deviate from this ideal behavior due to several non-ideal effects. One of the major causes of performance degradation in the SC integrators is the incomplete transfer of charge. This non-ideal effect is a consequence of the operational amplifier non-idealities, namely finite gain and bandwidth, slew rate and saturation voltages. Fig. 1 shows the model of the integrator including all the non-idealities, which will be considered in detail in the more complex circuits discussed in the next paragraphs. The MATLAB function that implements the Slew Rate, actually comprises Eqns. (4), (5), (6), (7) and (8).

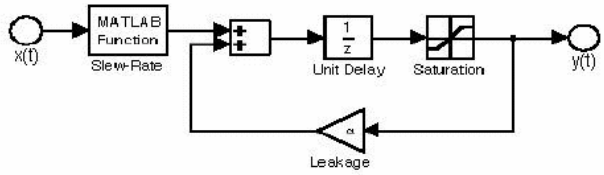

Fig. 1 Simulink model of an SC integrator

The op-amp of the integrator described by Eqn. (1) is ideal. However, the gain of any op-amp is finite and this causes a first limit. The effect is an integrator "leakage": only a fraction of the previous output of the integrator is added to each new input sample. The transfer function of the integrator with leakage becomes

$$H(z)=\frac{z^{-1}}{1-\alpha z^{-1}}. \qquad (2)$$

The dc gain H_0 becomes therefore:

$$H_0=\frac{1}{1-\alpha}. \qquad (3)$$

The effect of the finite bandwidth and the slew-rate are related to each other and may be interpreted as a non-linear gain. The evolution of the output node during the nth integration period is governed by

$$v_0(t)=v_0(nT-T)+\alpha V_s\left(1-e^{-\frac{t}{\tau}}\right),\ nT-\frac{T}{2}<t<nT, \qquad (4)$$

where $V_S = V_{in}(nT - T / 2)$, is the integrator leakage and is the time constant of the integrator (*GBW* is the unity gain frequency of the operational amplifier when loaded by C_f). The slope of this curve reaches its maximum value when *t=0*, resulting in

$$\frac{d}{dt}v_0(t)\Big|_{\max}=\alpha\frac{V_s}{\tau}. \qquad (5)$$

We must consider now two separate cases:

1. The value specified by Eqn. (5) is lower than the operational amplifier slew-rate, *SR*. In this case there is not slew-rate limitation and the evolution of v_0 conforms Eqn. (4).
2. The value specified by Eqn. (5) is larger than *SR*. In this case, the operational amplifier is in slewing and, therefore, the first part of the temporal evolution of v_0 ($t < t_0$) is linear with slope *SR*. The following equations hold (assuming $t_0 < T$

$$t < t_d,\ \ v_0(t) = v_0(nT - T) + SRt, \tag{6}$$

$$t > t_d.\ v_0(t) = v_0(t_0) + (\alpha V_s - SRt_0)\left(1 - e^{-\frac{t-t_0}{\tau}}\right). \tag{7}$$

Imposing the condition for the continuity of the derivatives of Eqn. (6) and Eqn. (7) in t_0, we obtain

$$t_0 = \frac{\alpha V_s}{SR} - \tau. \tag{8}$$

If $t_0 > T$ only Eqn. (6) holds. The MATLAB function in Fig.1 implements the above equations to calculate the value reached by v_0 (t) at time T, which will be different from V_0 due to the gain, bandwidth and slew-rate limitations of the operational amplifier.

The saturation voltages of the operational amplifier are modeled using the saturation block inside the feedback loop of the integrator, as shown in Fig.1.

Comparator

A widely used configuration of comparator comprises the cascade of a preamplifier and a latch (with hysteresis). The input signal can vary significantly; then, the amplifier can operate in the linear or the overdrive region of operation. Therefore, as we have for the integrator discussed above, the behavioral model should be able to determine the region of operation and to apply the proper behavioral model. The final stage of the comparator is a latch with hysteresis; the hysteresis is included to account for the metastability error due to signals whose magnitude is very close to the comparator reference. Additionally, fixed offset voltage and random offset voltage must be added to the input to model the offset and any time dependent spur signals.

The block diagram of the SIMULINK model is given in Fig.2. Several SIMULINK blocks in Fig. 2 depict an operation similar to the one represented by the MATLAB function in Fig. 1.

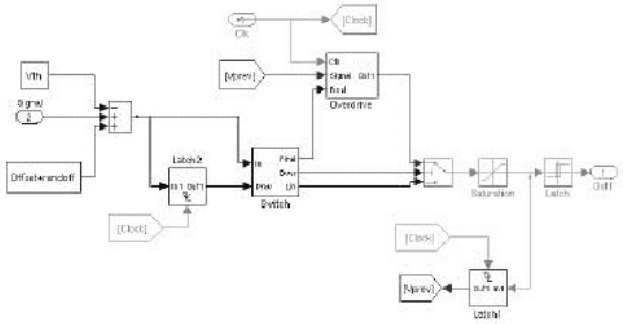

Fig 2. SIMULINK model of a comparator

Capacitive and resistive array

The limit caused by an array must be studied at the circuit level and the effect must be then translated into a behavioral description. The static mismatch of nominally equal elements can be described by the use of random numbers with a given variance stored on an array. More complex is the modeling of dynamic limitations. Let us consider, for example, the dynamic evolution of the string of equal resistances that a flash converter uses.

The auto-zero and the parasitic capacitance of all the comparators are pre-charged to the input voltage and then, they are connected to the intermediate nodes of the resistive string. The transient voltage of the intermediate nodes leads to diagrams similar to the ones in Fig. 2. If the time allowed by the phase used is not enough the voltages of the intermediate nodes do not reach the expected value. A similar limit occurs during the next complementary phase, when the auto-zero capacitor is charged to the new input voltage. Therefore, the combination of the two limits leads to a memory error.

Simulation Environment

The simulation environment uses the MATLAB-SIMULINK platform but it can be easily translated in other simulation frameworks. The environment is flexible enough to cover a wide range of specifications and applications. It has a main menu GUI, as shown in Fig 3. It covers popular architectures like flash converter, pipeline and sigma-delta are defined in hierarchical SIMULINK descriptions. The system also incorporates a library of behavioral models that represent the basic blocks used for building data converter architectures. The designer can build with these blocks a custom architecture and try new ideas. Each block has a set of parameters that define its specifications so the designer is able to see the effect of each parameter on the whole system performance.

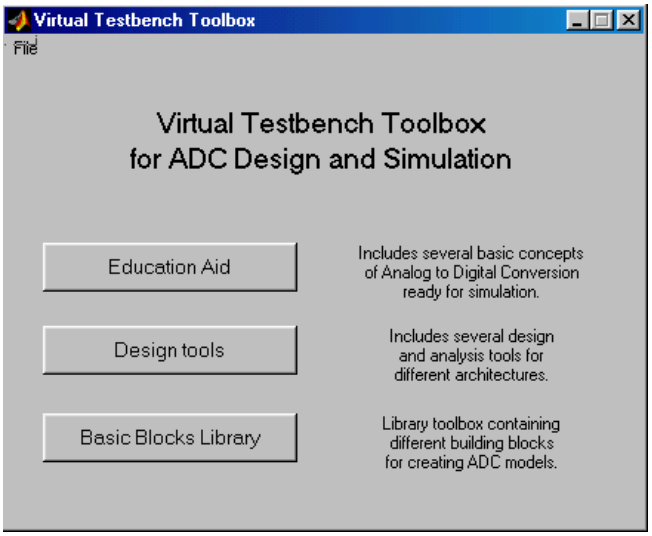

Fig 3. Main menu GUI

These parameters represent non-idealities that have been included in the models. Dialog masks permit a quick specification of parameters. The behavioral models of each block have been validated with SPICE simulations.

An example of basic block included in the library is the switched capacitor integrator used in sigma-delta modulator. The model accounts for the finite gain, bandwidth and slew-rate of the op-amp, linearity of the capacitors used, and permits to simulate the effect of white noise sources. Fig.4 shows the GUI of the block library of the environment.

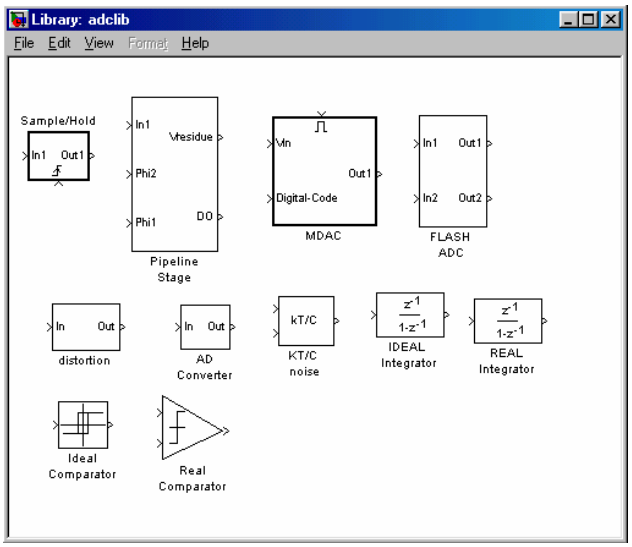

Fig 4. GUI for basic blocks library

Implied in the design activity is the analysis of the data converter performance. For this analysis several functions were developed to calculate static and dynamic performance.
Among them, the tool permits to estimate the INL and the DNL; it is also possible to evaluate the SNR, the SFDR and other dynamic parameters. Moreover, the environment includes a number of options for plotting diagrams, running parametric analysis, where a certain performance can be studied for particular parameter variations.

Beyond the above-described features, the simulation environment provides an additional benefit: the environment includes a specific section for educational purposes. The section provides specific examples and design guidelines for illustrating basic concepts of data converter operations. The educational aid has available the same analysis tools used for design. Several exercises proposed allow the user to study the features and limitations of the architectures included. The designer can also use the predefined architectures as a starting point and then modify it to his particular needs. The education aid section is illustrated in Fig. 5, 6, 7 and 8. Fig. 5 shows the menu that permits the student to study basic concepts like the effect of ideal and non-ideal sampling and the spectrum of the quantization noise for different input frequency-sampling ratios.

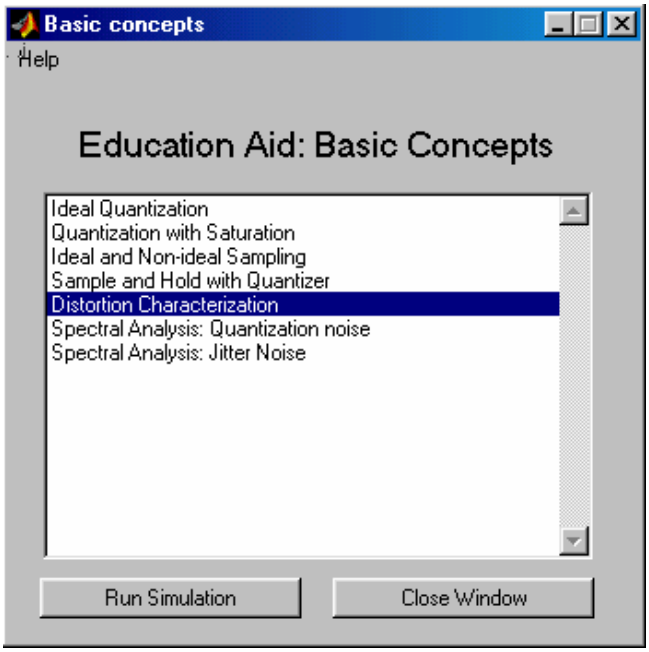

Fig 5. Education aid GUI

Fig. 6 shows the simulation control menu. It includes a Help button and buttons for setting the simulation parameters. Fig. 7 shows the graphic interface for setting parameters as numerical inputs or cursor-like. Fig. 8 shows the block diagram of the example selected with the addition of the run and the help button.

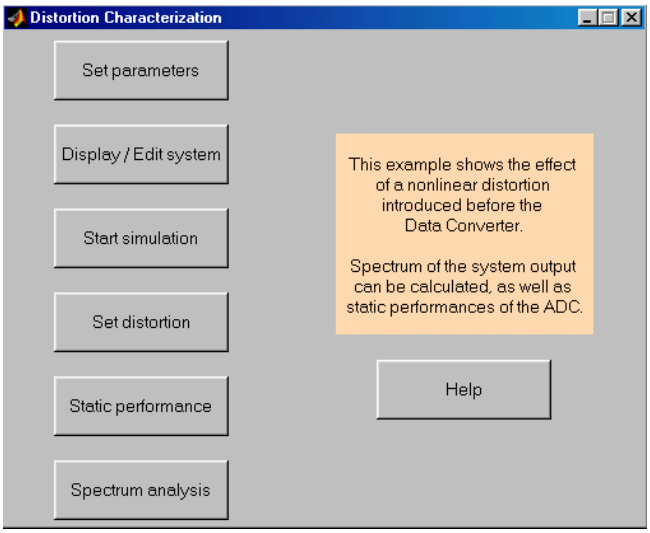

Fig 6. Main menu GUI for an education aid example

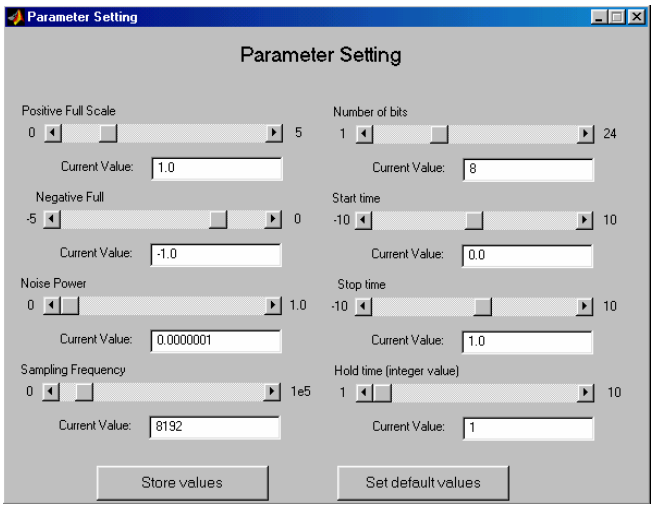

Fig 7. GUI for parameter setting

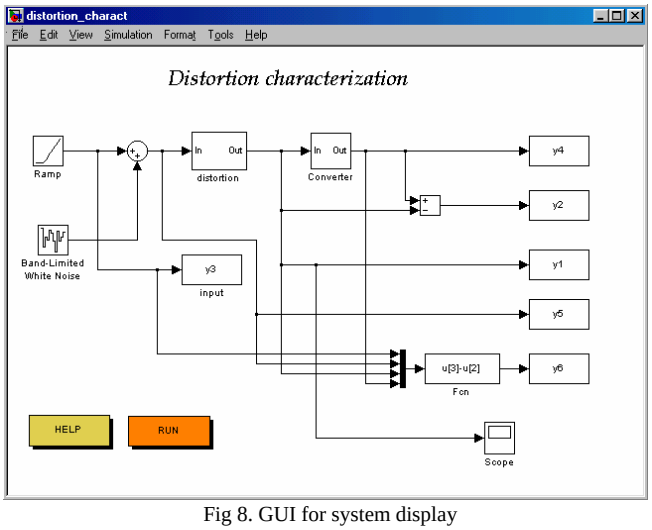

Fig 8. GUI for system display

Finally the simulation environment has a section where the designer can choose among popular data conversion algorithms. Set the specifications requirements according to his needs and then study the requirements for each individual block that he must be able to satisfy in order to achieve the desired performance. This section is showed in Fig. 9 and 10. Fig. 9 shows the menu where the user can select the desired conversion algorithm. By selecting any of these options, the user is presented with an input menu where specifications of the particular conversion algorithm can be entered. Fig. 10 shows an example where the user has selected the pipeline architecture and entered in the specifications that 3 stages where desired. The software synthesizes a block diagram of the converter with the desired specifications.

We assume some specification requirements for the data converter. An ideal architecture achieves better performance than a real implementation. Therefore, the specification requirements should leave some margin to account for the non-idealities. The designer should use this "budget" properly. The non-critical parameters will utilize a little of the budget, so that the major part of it can be devoted to the real limitations. For example, in a medium resolution high-speed converter, the gain is not very important, the offset won't be a problem (if the *dc* response is not relevant) but the gain-bandwidth and, even more the slew-rate, are very critical. The results of the behavioral simulations quantify the relevance of the above mentioned limitations and permit a proper "budget" assignment for an optimum design trade-off.

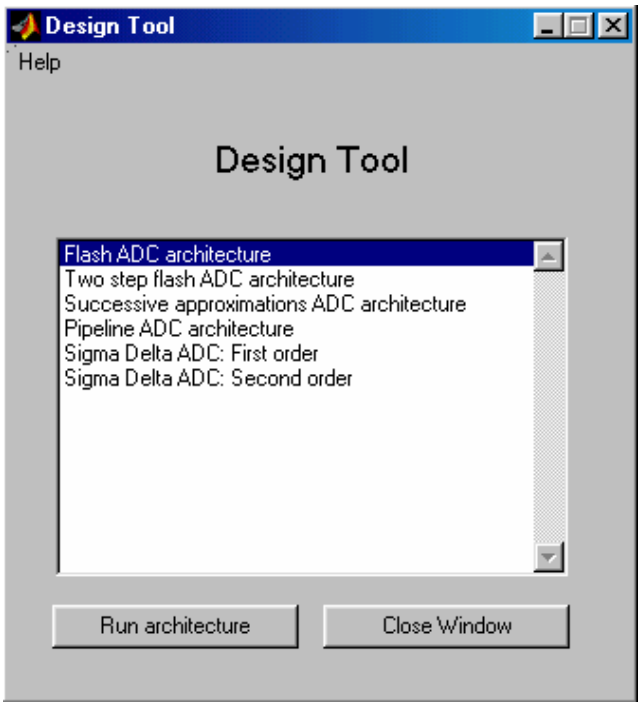

Fig 9. GUI for design tools

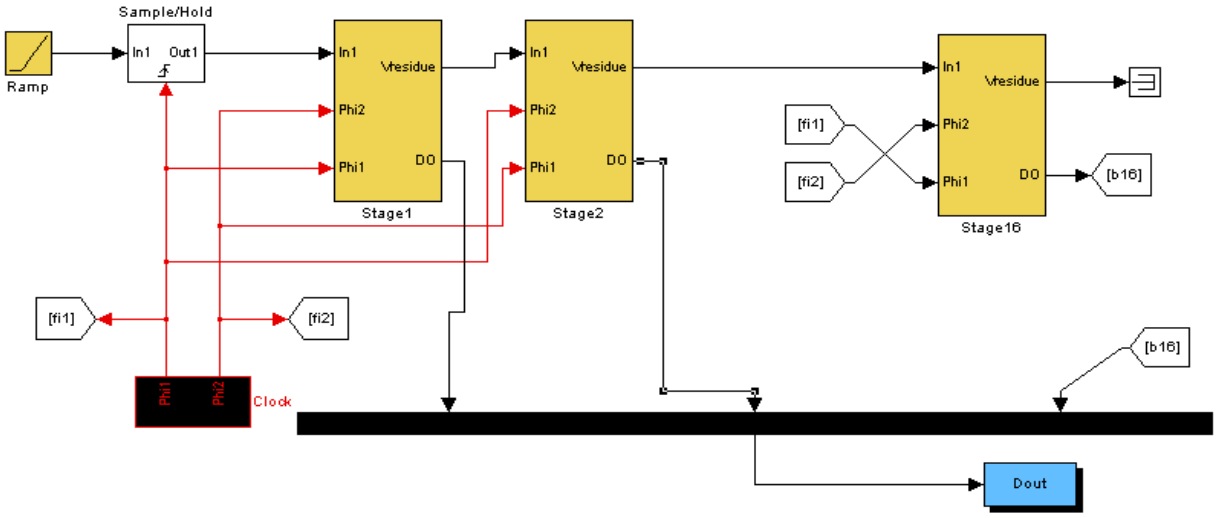

Fig 10. Synthesized 3 stage pipeline behavioral model

Conclusions

In summary, the developed user-friendly environment allows the user to design a data converter and define its basic block specifications at system level. By using behavioral models, it significantly reduces the simulation time and, for educational purposes, with behavioral description it helps in better understanding architectures and the limitations associated to the real blocks used.

References

[1] *F. Maloberti, P. Estrada, P. Malcovati, A. Valero*, "Behavioral Modeling and Simulations of Data Converters" Journal of the International Measurement Confederation IMEKO, *June. 2001*.

[2] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez, J. L. Huertas, "Modeling OpAmp-Induced Harmonic Distortion for Switched-Capacitor Modulator Design", *Proceedings of ISCAS '94*, vol. 5, pp. 445-448, London, UK, 1994.

[3] S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto and F. Maloberti", Modeling Sigma-Delta Modulator Non-Idealities in SIMULINK", *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '99)*, 2, Orlando, USA, pp. 384-387, 1999.

[4] *Gielen, G. E. – Franca, J. E.* "CAD tools for data converter design: an overview" IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, page(s): 77 - 89 Feb. 1996 Volume: 43 Issue: 2