Simulation-based High-Level Synthesis of Nyquist-Rate Data Converters using MATLAB/SIMULINK

Jesús Ruiz-Amaya, José M. de la Rosa*, Manuel Delgado-Restituto and Angel Rodríguez-Vázquez

Instituto de Microelectrónica de Sevilla, IMSE-CNM (CSIC)
Edificio CICA-CNM, Avda Reina Mercedes s/n, 41012-Sevilla, SPAIN

ABSTRACT

This paper presents a toolbox for the simulation, optimization and high-level synthesis of Nyquist-rate Analog-to-Digital (A/D) and Digital-to-Analog (D/A) Converters in MATLAB®. The embedded simulator uses SIMULINK® C-coded S-functions to model all required subcircuits including their main error mechanisms. This approach allows to drastically speed up the simulation CPU-time up to 2 orders of magnitude as compared with previous approaches – based on the use of SIMULINK® elementary blocks. Moreover, S-functions are more suitable for implementing a more detailed description of the circuit. For all subcircuits, the accuracy of the behavioral models has been verified by electrical simulation using HSPICE. For synthesis purposes, the simulator is used for performance evaluation and combined with an hybrid optimizer for design parameter selection. The optimizer combines adaptive statistical optimization algorithm inspired in simulated annealing with a design-oriented formulation of the cost function. It has been integrated in the MATLAB/SIMULINK® platform by using the MATLAB® engine library, so that the optimization core runs in background while MATLAB® acts as a computation engine. The implementation on the MATLAB® platform brings numerous advantages in terms of signal processing, high flexibility for tool expansion and simulation with other electronic subsystems. Additionally, the presented toolbox comprises a friendly graphical user interface to allow the designer to browse through all steps of the simulation, synthesis and post-processing of results. In order to illustrate the capabilities of the toolbox, a 0.13μm CMOS 12-bit@80MS/s analog front-end for broadband power line communications, made up of a pipeline ADC and a current steering DAC, is synthesized and high-level sized. Different experiments show the effectiveness of the proposed methodology.

Keywords: Analog-to-digital converters, digital-to-analog converters, optimization, behavioral modeling and simulation.

1. INTRODUCTION

The exponential increase of the capabilities of digital CMOS circuits – fuelled by the evolution of process technologies towards deep submicron – is prompting the integration of complete electronic systems onto a single chip. In such Systems-on-Chip (SoC), most of the signal processing is carried out by digital circuitry, whereas the role of analog circuits basically reduces to implement the necessary signal conditioning and data conversion interfaces [1,2]. In spite of this apparently minor role, the design of high-performance analog circuitry (usually, in adverse digital-oriented technologies) most often represents an important bottleneck for a short time-to-market deployment [3].

This problem is aggravated in modern telecommunication applications, like Very high-rate Digital Subscriber Line (VDSL) and Power Line Communication (PLC), where data converters targeting 12-14bit resolution at conversion rates of 40-80 MSamples/second (MS/s) are needed [1]. Although such data rates are easily achievable with flash or folding/interpolation Analog-to-Digital Converters (ADCs), their area and power consumptions become so significant at resolutions beyond 10 bit, that makes their deployment not competitive at least for SoC applications [4,5]. On the other hand, the use of ΣΔ modulator topologies is neither a viable solution for high signal bandwidths (beyond 15MHz) because of the prohibitive sampling frequencies which are required to achieve medium-high resolution [6,7].

In this scenario, pipeline ADCs have demonstrated to be a good alternative for interfaces requiring medium-high resolution at video-range conversion rates and beyond [8,9], whereas current-steering architectures are normally used to implement the D/A Converter (DAC) [1].

* jrosa@imse.cnm.es; phone +34955056666; fax +34955056686; www.imse.cnm.es

VLSI Circuits and Systems II, edited by José Fco. López, Francisco V. Fernández, José María López-Villegas, José M. de la Rosa, Proceedings of SPIE Vol. 5837 (SPIE, Bellingham, WA, 2005) 0277-786X/05/$15 · doi: 10.1117/12.607832

This has motivated the interest for CAD tools which can optimize and shorten the synthesis procedure of Nyquist-rate data converters, especially those based on pipeline architectures [10-14]. Most of them are based on an iterative optimization procedure in which the design problem is translated into a cost function minimization problem that can be evaluated through numerical methods. Evaluation of the cost function is normally performed by means of equations [11,12-14], so that very short computation times are obtained. As a drawback, this approach results in closed tools because equations must be changed every time the topology is changed.

This paper aims at palliating this problem by using simulation instead of equations for cost function evaluation. To this end, a complete toolbox for the high-level synthesis of Nyquist-rate data converters has been developed in the MATLAB® environment [15]. The embedded simulator uses SIMULINK® C-coded S-functions [16] to model all required subcircuits including their main non-idealities. This approach considerably reduces computational costs as compared to using standard library blocks as in [13]. For all subcircuits, the accuracy of the behavioral models has been verified by electrical simulations using HSPICE. Additionally, the toolbox includes an improved version of an efficient hybrid optimizer which uses statistical techniques for design space exploration and deterministic techniques for fine tuning[17]. Other important features of the platform are a friendly Graphical User Interface (GUI), wide signal processing capabilities and high flexibility for tool expansion [15].

The proposed toolbox covers different converter topologies such as full flash and pipeline ADCs and current steering DACs. As a case study, a pipeline ADC and a current steering DAC intended for a 12-bit@80MS/s PLC analog front-end are synthesized and high-level sized in a 0.13µm standard CMOS technology.

2. PROPOSED SYNTHESIS TOOLBOX

The proposed high-level synthesis toolbox is based on the combination of a hybrid optimizer for design parameter selection and a time-domain behavioral simulator for performance evaluation. Both tools are integrated in the MATLAB/ SIMULINK® environment as described below.

2.1 Optimization engine

For synthesis purposes, deterministic optimization methods, like those available in the MATLAB® standard distribution [15], are not suitable because initially designers may have little or no idea of an appropriate design point. Therefore, the optimization procedure is quickly trapped in a local minimum. For that reason, we developed an optimizer which combines an adaptive statistical optimization algorithm inspired in simulated annealing (local minima of the cost function can then be avoided) with a design-oriented formulation of the cost function (which accounts for the modulator performances). Moreover, an integrated approach is addressed: statistical techniques are applied for wide design space exploration whereas deterministic techniques are used for fine-tuning of best solutions found by the previous techniques. Unlike conventional simulated annealing procedures, in which the control parameter – commonly named temperature – follows a predefined temporal evolution pattern, the implemented global optimization algorithm dynamically adapts this temperature to approximate a predefined evolution pattern of the acceptance ratio (accepted movements / total number of iterations). This idea prevents excessively high temperatures which will make convergence difficult and inappropriately low temperatures which can make the algorithm to stuck on a local minimum. The amplitude of parameter movements through the design space is also synchronized with the temperature for improved convergence.

This optimizer has been integrated in the MATLAB®/SIMULINK® platform by using the MATLAB® engine library [15], so that the optimization core runs in background while MATLAB® acts as a computation engine. Fig.1 shows the flow diagram of the optimizer where starting from an ADC topology, e.g., an arbitrary ADC whose design parameters (building block specifications, resolution per stage, etc.) are not known and arbitrary initial conditions, a set of design parameter perturbations is generated. With the new design parameters, a set of simulations are done to evaluate the circuit performance. From the simulation results, it automatically builds a cost function (that has to be minimized). The type and value of the perturbations as well as the iteration acceptance or rejection criteria depend on the selected optimization method. The optimization process is divided into two steps. The first step explores the design space by dividing it into a multidimensional coarse grid, resulting in a mesh of hypercubes (*main optimization*). A statistical method is usually applied in this step. Once the optimum hypercube has been obtained, a final optimization – using a deterministic method – is performed inside this hypercube (*local optimization*).

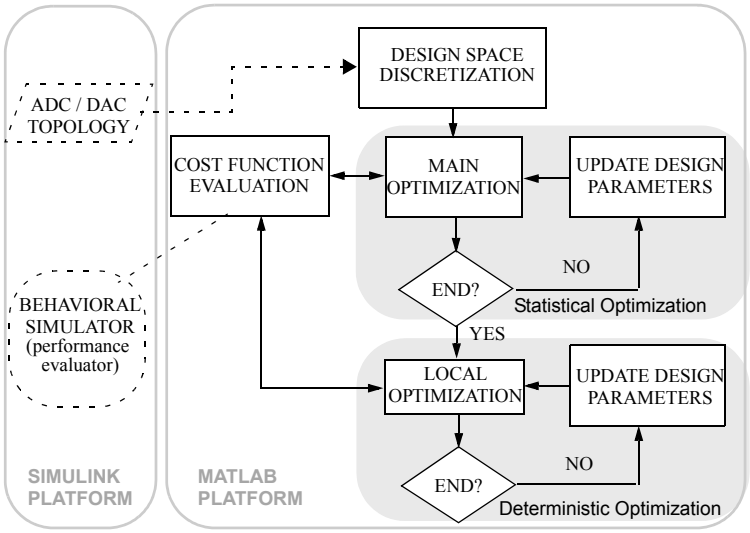

Figure 1. Operation flow of the optimization core.

The optimization core is very flexible, in so far as the cost function formulation is very versatile: multiple targets with several weights, constraints, dependent variables, and logarithmic grids are permitted. This optimization procedure has been extensively tested with design problems of ΣΔ modulators involving behavioral simulators as well as electrical simulators [17].

2.2 SIMULINK-based behavioral simulator

The iterative nature of the optimization procedure requires a very efficient mechanism for performance evaluation. Event-driven behavioral simulation implemented in the SIMULINK® platform is used in the proposed synthesis toolbox. This technique enables very efficient analysis while providing high accuracy levels [3].

Behavioral modeling using SIMULINK® was first applied to pipeline ADCs by Bilhan et al. [13]. Although very intuitive, the implementation of the behavioral models of each basic building block requires several sets of elementary SIMULINK® blocks. This means a penalty in computation time which may become critical in an optimization-based synthesis process in which hundreds or thousands of simulations must be executed.

To overcome this problem, in the simulator in this paper, behavioral models have been incorporated in the SIMULINK® environment by using C-coded S-functions [16]. This approach allows to drastically speed up the simulation CPU-time[††] (up to 2 orders of magnitude) as compared to previous approaches [13]. Moreover, S-functions are more suitable for implementing a more detailed description of the different subcircuits, taking into account all critical non-linear error mechanisms as will be detailed in Section 3.

2.3 Implementation in the MATLAB® environment

The proposed tool has been conceived as a MATLAB® toolbox for the simulation and synthesis of Nyquist-rate data converters, including flash, pipeline ADCs and current-steering DACs. Fig.2 shows some parts of the toolbox comprising a GUI to allow the designer to browse through all steps of the simulation, synthesis and post-processing of results. By using this GUI, the designer can either open an existing ADC or DAC architecture or create a new one in the SIMULINK® platform by connecting the building-blocks available in the toolbox. After a simulation is done, different figures such as output spectrum, in-band noise power, static non-linearity, harmonic distortion, etc., can be computed through the analysis/data processing menu. High-level synthesis is started from the synthesis menu, where constraints, performance speci-

[††] For instance, a 32738-samples simulation of a typical pipeline ADC takes 2-3 seconds. All simulations shown in this paper were done using a PC with an AMD XP2400 CPU@2GHz @512MB-RAM.

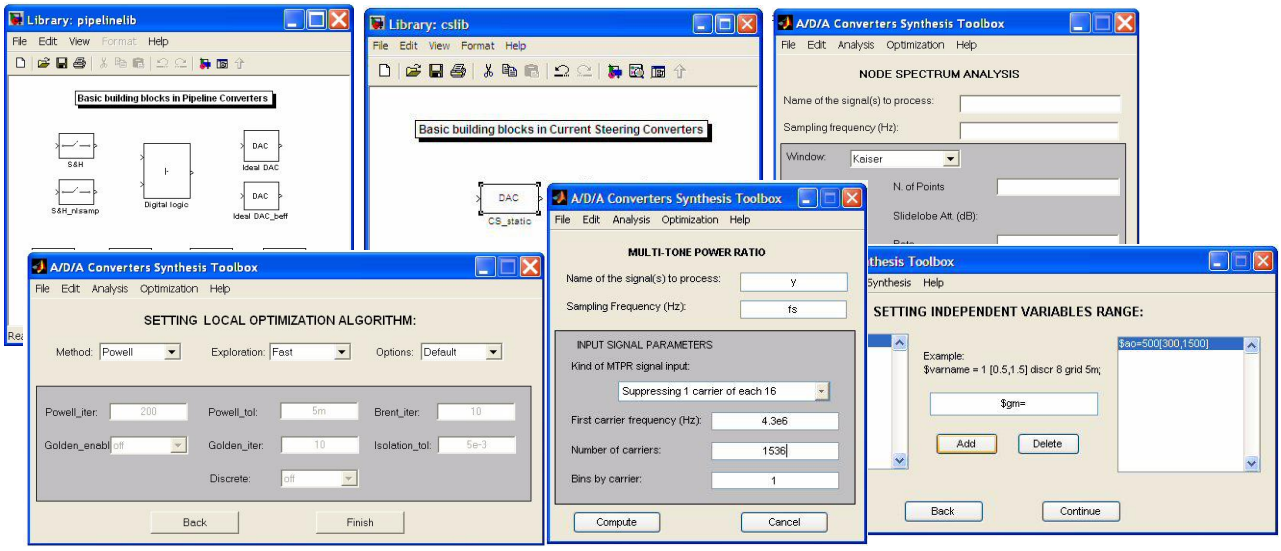

Figure 2. Illustrating the data converter toolbox.

fications, design parameters, optimization algorithms, etc., can be specified. Then, the optimization core starts the exploration of the design space to find out the optimum solution by using the simulation results for performance evaluation.

3. BEHAVIORAL MODELING OF NYQUIST-RATE DATA CONVERTERS USING SIMULINK C-CODED S-FUNCTIONS

As described above, behavioral modeling is a critical part of the proposed synthesis toolbox. However, a detailed description of the model of each embedded ADC/DAC subcircuit is beyond the scope of this paper. Instead of that, we will focus on the case of pipeline ADCs and current steering DACs as an application.

3.1 Modeling of pipeline ADC building blocks

Fig.3(a) shows the conceptual block diagram of a generic pipeline ADC, consisting of an arbitrary cascade of k stages and a Sampled-and-Hold (S/H) circuit at the front [1,2,11]. Each stage resolves partial code words of length n_j, $j = 1, \ldots, k$, which are all re-ordered and combined at the digital correction block to obtain the N bit output of the converter. The inner

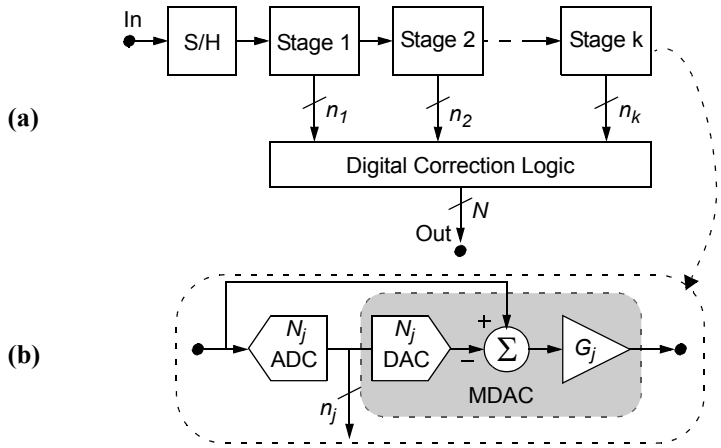

Figure 3. Generic pipeline ADC architecture. (a) Conceptual block diagram; (b) Single stage.

structure of a pipelined stage comprises four blocks, as illustrated in Fig.3(b): a flash sub-ADC with $N_j \le 2^{n_j}$ output codes, a sub-DAC with N_j output levels, a substractor, and a S/H residue amplifier with gain G_j. The latter three blocks are implemented in practice by a single subcircuit which is often referred to as Multiplying DAC (MDAC). All these blocks have been modelled in the proposed toolbox as described below.

3.1.1 S/H circuit

Fig.4(a)[†††] shows the conceptual schematic of a typical S/H block topology which operates with two non-overlapped clock phases. Its model in the proposed toolbox includes the most critical error mechanisms which are computed according to the flow diagram in Fig.5. The flow graph has two branches corresponding to the two clock phases. During the sampling phase, the input-equivalent thermal noise, (Δv_n), is calculated and added to the voltage stored at the sampling capacitor C_s. This is computed taking into account the non-linear switch on-resistance effects (r_{on}). Next, an iterative procedure[††††] is started to calculate the output voltage V_o by solving the equivalent circuit of Fig.4(b), which models the effects of finite and non-linear opamp DC-gain ($A_v = G_m(V_a)/g_o$), opamp offset (E_{os}), non-linear sampling capacitor, transient response (comprising both linear incomplete settling and slew-rate limitation), parasitic capacitances (C_p, C_b, C_{load}), output range limitations and charge injection error. During the hold phase, a similar procedure is applied to solve the equivalent circuit shown in Fig.4(c).

As the value of state signals are important only at the end of each clock phase, a set of finite difference equations have

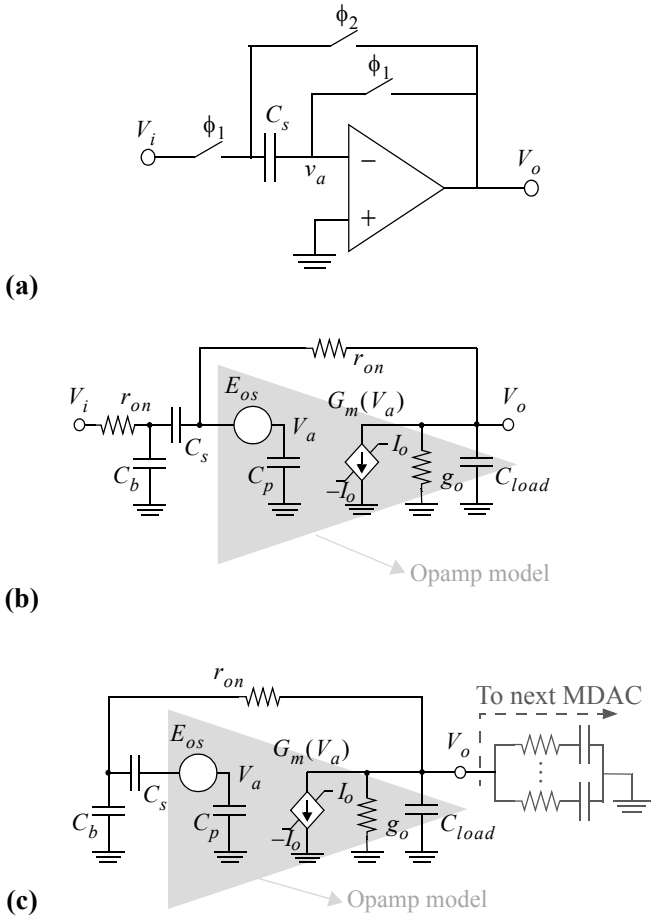

Figure 4. S/H (a) schematic. Equivalent circuit in (b) sampling phase and (c) hold phase.

[†††] For simplicity, schematics are shown in its single-ended version, although actually the fully-differential structures have been modelled.

been generated and codified using C-compiled S-functions in order to describe the operation of real S/H circuits in the presented toolbox. For this purpose, SIMULINK® provides different S-function templates which can accommodate the C-coded computation model of both DT and CT systems. These templates are composed of several routines that perform different tasks required at each simulation stage [16]. Among others, these tasks include: variable initialization, computation of output variables, update of state variables, etc. For illustration purposes, Fig. 6(a) shows some significant sections of the S-function file associated to the S/H circuit of Fig.5. It includes model parameters, clock phase diagram, computation model code, etc. Once the S-function has been created, it is compiled by using the *mex* utility provided by MATLAB® [15]. The resulting object files are dynamically linked into SIMULINK® when needed. The block model is incorporated into the SIMULINK® environment by using the S-function block of the SIMULINK® libraries [16]. Fig.6(b) illustrates this process for the S/H circuit. A block diagram containing the S-function block is created including the input/output pins. The dialogue box is used to specify the name of the underlying S-function. In addition, model parameters are also included in this box, which can be used to modify the parameter values.

As an example of the accuracy of the behavioral model, Fig.7 compares the transient response for a constant input voltage by using HSPICE and our model showing a good agreement.

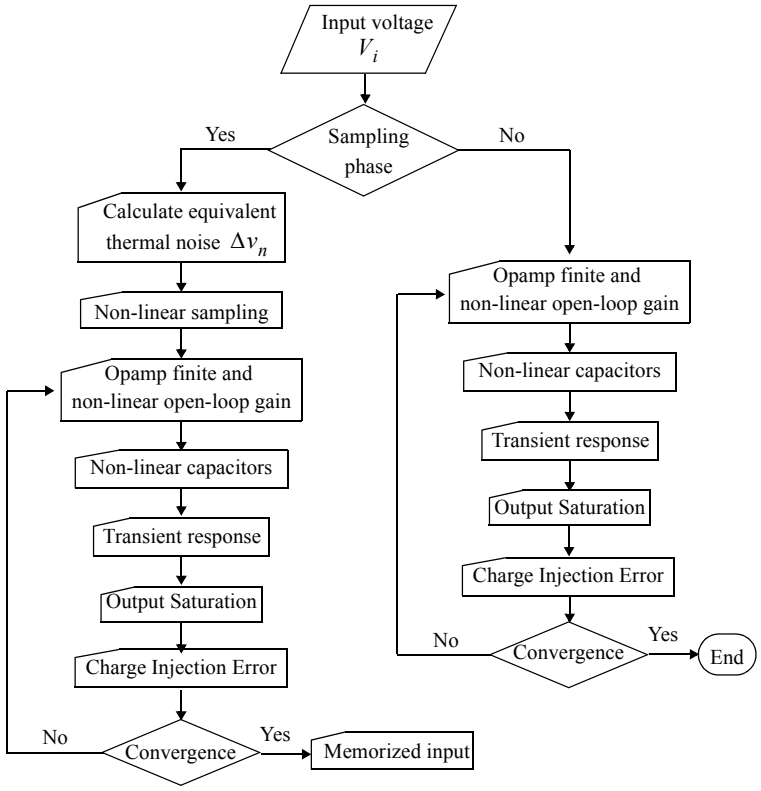

Figure 5. Flow diagram of the S/H model.

†††† The convergence criterion used in the iterative procedure of the behavioral models is: $\text{abs}[(\text{New\_parameter\_value} - \text{Old\_parameter\_value})/\text{New\_parameter\_value}] < \text{thrs}$, where thrs is the threshold value chosen for convergence (normally $\text{thrs} = 0.01$), abs(x) stands for the absolute value of x, and New_param_value and Old_param_value are respectively the old and new values of the parameter to be solved – for instance the DC gain in the flow diagram of Fig.5. Using this criterion, convergence is reached normally in 3 or 4 iterations, which does not result in excessively costly CPU time.

(a)

```
#define S_FUNCTION_NAME sh
.....
#define phi(S) ssGetSFcnParam(S,0)
#define cs(S)  ssGetSFcnParam(S,1)
#define Ts(S)  ssGetSFcnParam(S,2)
.....
static void mdlInitializeSampleTimes(SimStruct *S)
{
   double TS;
   TS = *mxGetPr(ts(S));
   ssSetSampleTime(S, 0, TS/2);
   ssSetOffsetTime(S, 0, 0);
}
.....
static void mdlOutputs(SimStruct *S, int_T tid)
{.....
if (phi==1){
   Va=set_samp(ron,gm,io,*vi[0], ceqs, cload,...);
   vnoise=cnoise(ron,gm,opnoise,...);
   work[0]=(*vi[0]+vnoise-Va);}
else{
   Va=set_amp(ron,gm,io,work[0],ceqi,...);
   Vo=foutput(Va,work[0],Avnl...);}
```

(b)

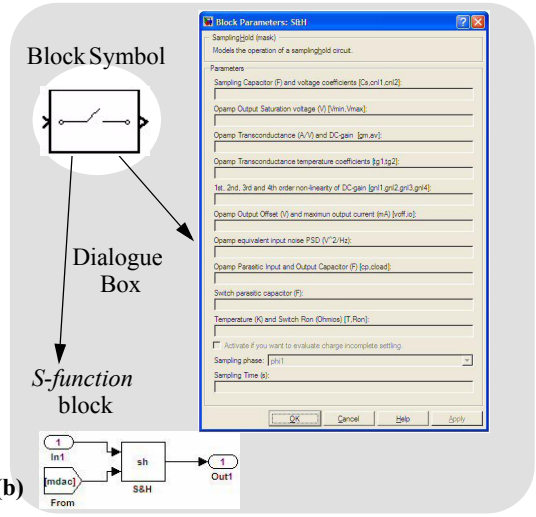

Figure 6. S-function of the S/H circuit in Fig. 5: (a) Excerpt of S-function code; (d) S-function block.

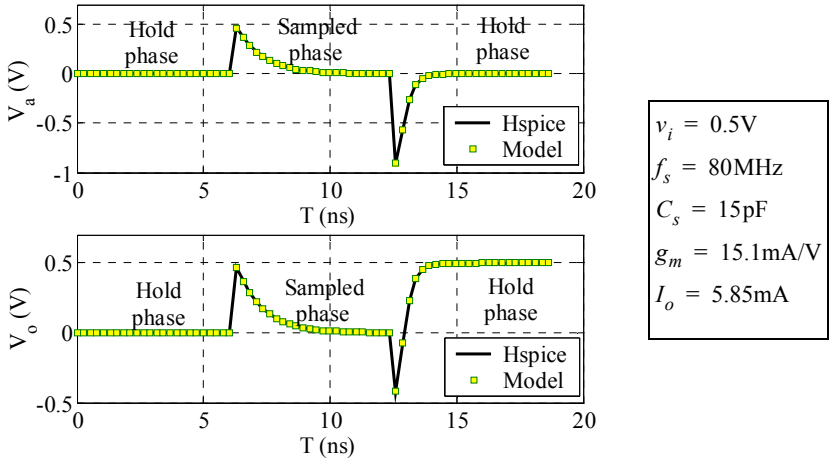

Figure 7. Transient response of the S/H: comparison between HSPICE and our behavioral model.

3.1.2 MDAC

Fig.8(a) shows the conceptual schematic of a MDAC. Critical aspects affecting linearity and transient response have been considered: capacitance mismatch and non-linearities ($C_u = C_{uo}(1 + cnl_1 v + cnl_2 v^2)$), finite switch -on resistances (r_{on}) and opamp errors (offset (v_{off}), finite and non-linear DC gain, thermal noise, incomplete settling and slew-rate). The flow diagram of the operation of the MDAC is similar to the one of the S/H, but the equivalent circuits used to evaluate the internal nodes are the ones which are shown in Fig.8(b) and Fig.8(c). For the opamp, a two-pole model (g_m, C_{load}, g_o, g_{mh}, C_h, g_{oh}) using Miller compensation (C_c) has been developed. This model matches very well with HSPICE predictions as shown in Fig.9.

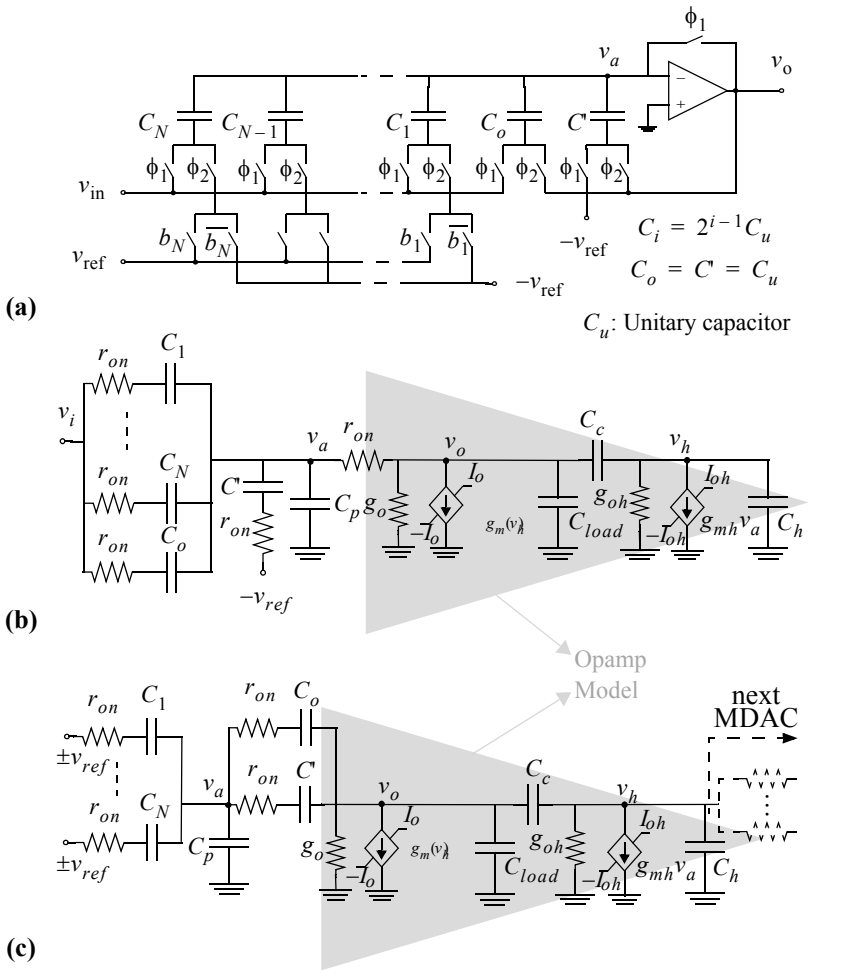

Figure 8. MDAC (a) schematic. Equivalent circuit in (b) sampling phase and (c) residue amplification phase. (Opamp offset not shown for simplicity).

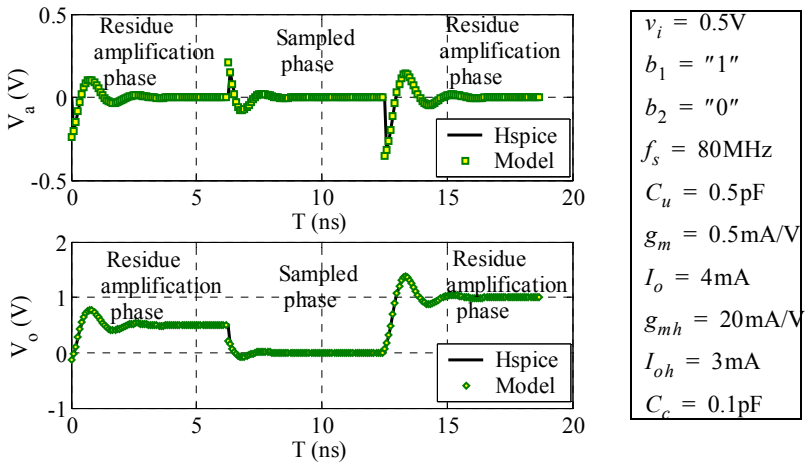

Figure 9. Transient response of a 2-bit MDAC considering a constant input voltage: comparison between HSPICE and two-pole behavioral model.

3.2 Modeling of current-steering DACs

The basic block diagram of a segmented current steering DAC is shown in Fig.10(a), where the input *N*-bits are split in *b* Least Significant Bits (LSB's), which steer a binary weigthed array of current sources – binary segment –, and $(t = N - b)$ Most Significant Bits (MSB's), which are thermometer-wise decoded, steering a unary array of current sources – thermometer segment –. The simplest topology implementing the current cell consists of a single MOS transistor biased with constant gate-source voltage and operating in saturation. The switching is performed by means of MOS transistors operating in the saturation region with complementary activation gate signals. An alternative to the simple current cell is the cascode current cell shown in Fig.10(b). This circuit exhibits the same output resistance as a double-cascode structure when switches operate in saturation, which is enough for low-distortion applications[18]. The nominal operation of these current cells is affected by random errors (due to device mistmatches), systematic errors (finite output impedance, thermal gradients, edge effects, CMOS technology-related errors) and dynamic limitations[1,19]. These non-idealities have been analysed to develop a behavioral model of both topologies shown in Fig.11(a). The flow operation is as follows: firstly, the current source arrays (both binary and thermometer) are initialized and both random errors and gradient errors are considered. Secondly, the finite output impedance error is computed according to the input bits. Finally, the output voltage is calculated by evaluating the transient response of the equivalent circuit of the current cell in Fig.11(b), taking into account the switching sequence selected and the number of current sources which are switched on.

4. CASE STUDY: HIGH-LEVEL SIZING OF A 12-bit@80MS/s ANALOG FRONT-END FOR PLC

In order to illustrate the capabilities of the proposed toolbox for the simulation and synthesis of Nyquist-rate data converters, the high-level design of a 0.13µm CMOS 12-bit@80MS/s analog front-end for PLC will be described. The specifications are shown in Table 1. The objective consists of achieving those specifications with the minimum requirements

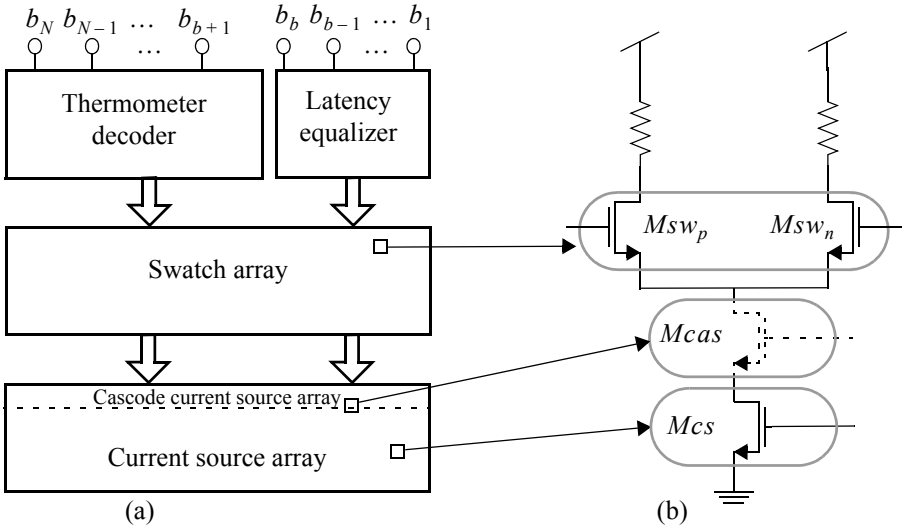

Figure 10. Current steering DAC: (a) Block diagram and (b) cascode current cell topology implementation.

Table 1: PLC analog front-end specifications

Pipeline ADC 12bit@80MS/s		Current Steering DAC 12bit@80MS/s	
Multi-Tone Power Ratio (*MTPR*)	≥ 56dB	Multi-Tone Power Ratio (*MTPR*)	≥ 56dB
Effective Number Of Bits (*ENOB*)	≥ 9.2 bits	Effective Number Of Bits (*ENOB*)	≥ 9.2 bits
Differential Input Range	2 Vp-p	Differential Input Range	1 Vp-p
Power Supply	3.3V	Power Supply	3.3V
		Output load	25Ω ‖ 20pF

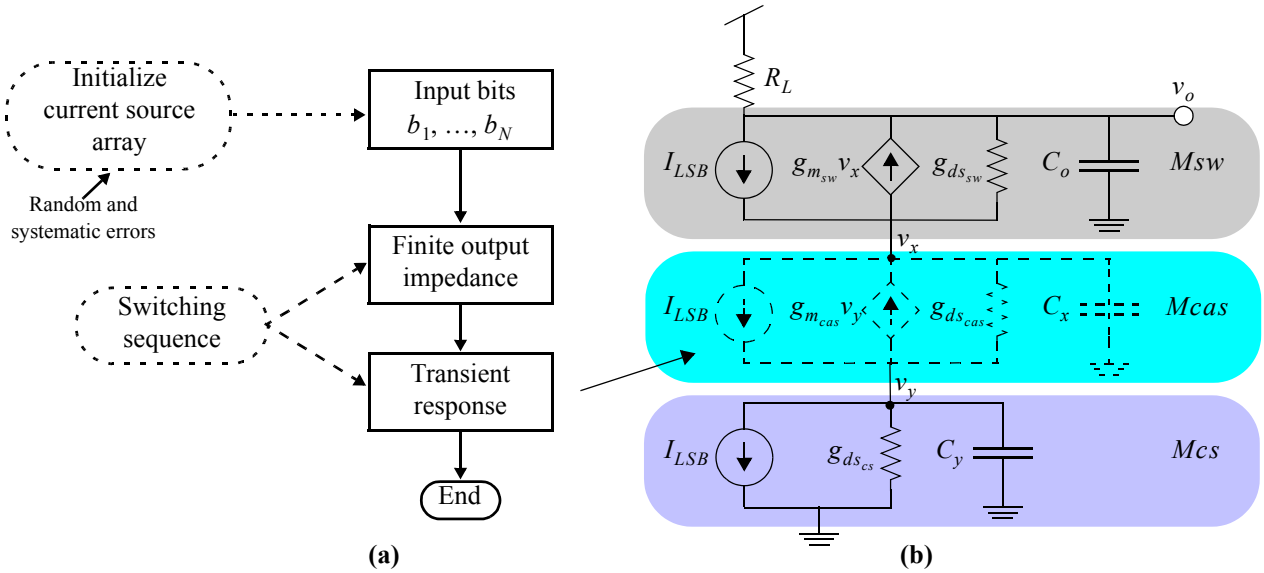

Figure 11. Behavioral model of current steering DAC's: (a) flow diagram and (b) equivalent circuit of the current cell.

in terms of power consumption and silicon area.

The design of the pipeline ADC is planned to be implemented without using calibration. For that reason, capacitor mismatch is a critical issue. In fact, this limitation forces us to optimize the capacitor sizes not only in terms of thermal noise and dynamics considerations but also in terms of minimum capacitance area needed to achieve the required mismatch. Another critical parameter considered in the optimization is the resolution per stage. Taking into account these factors, a wide exploration of several architectures has been carried out with the proposed synthesis toolbox. The optimum architecture was a 9-stage pipeline with 2 bits-per-stage except for the first one, which obtains 3 bits to improve the linearity of the ADC. The results of the high-level synthesis for the different stages as well as the requirements for the opamps are summarized in Table 2. The optimization procedure for a given architecture required about 3000 iterations of 1024 clock cycles taking about 20 minutes of CPU-time. The estimated power consumption is about 350 mW.

The synthesis toolbox has been also used to obtain the high-level specifications of the current steering DAC. A 8 thermometer – 4 binary-bit segmented architecture – has been selected, which guarantees a good accuracy-active area trade-off [20-22]. In order to reduce systematic gradient errors, the Q^2 Random Walk (Cong) switching sequence was selected [23]. On the other hand, random errors can be neglected if a relative standard deviation of the LSB current lower than 0.3% is achieved. Finally, a parametric analysis showed that a finite output impedance from DC to Nyquist frequency higher than 3MΩ is enough to guarantee a correct performance. Table 2 summarizes the high-level synthesis results. As an illustration, Fig.12 shows the output spectra for a single tone input at 38 MHz and an input DMT signal where 1536 tones are distributed from 4.3MHz to 34MHz, suppressing 8 tones of each 128.

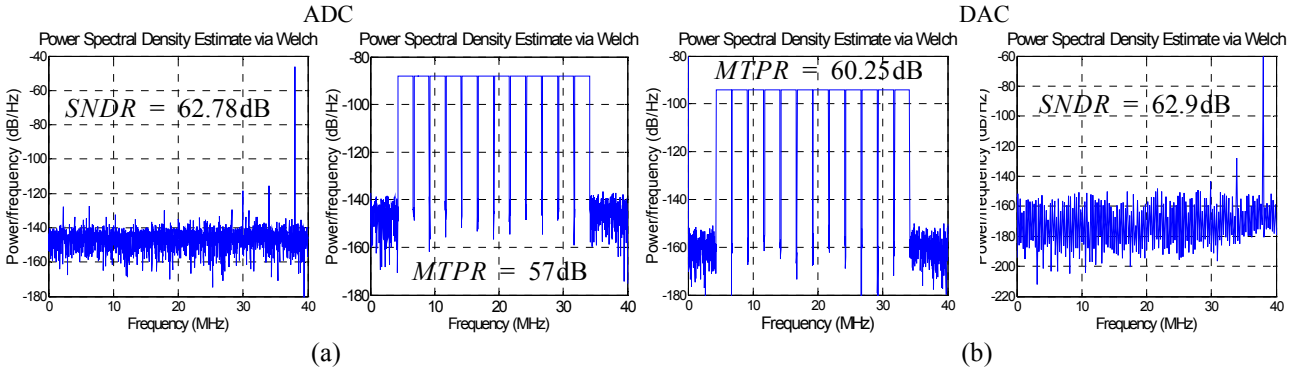

Figure 12. Power spectra for a single tone and a DMT test signal for the ADC (a) and the DAC (b).

Table 2: High-level synthesis results

Pipeline ADC			Stage 1	Stage 2-3	Stage 4-6	Stage 7-9
Flash Quantizer		Comparators Offset (mV)	< 20	<30	<30	<30
		Comparators Hysteresis (mV)	<30	<60	<60	<60
MDAC	Switch on-resistance (Ω)		<200	<200	<400	<400
	Opamp	Max. Eq. load (pF)	29.6	12.5	10.18	7.95
		Slew-Rate (V/μs)	>300	>309	>221	>100
		GB (MHz)	>240	>246	>176	>76
		DC-gain (dB)	>74.8	>68	>60	>54
		Noise PSD (nV/ $\sqrt{Hz}$ /)	<3	<5	<30	<140

Current Steering DAC	
Relative current standard deviation	<0.3%
Finite output impedance (from DC to Nyquist)	>3MΩ
Topology of the current cell	Cascode
Maximum gradient error	0.8%
Switching sequence	Q^2 Random Walk (Cong)

5. CONCLUSIONS

A complete MATLAB® toolbox for the high-level synthesis and verification of Nyquist-rate ADCs has been described. The combination of an efficient SIMULINK®-based time-domain behavioral simulator and an advanced statistical optimizer allows to efficiently map system-level specifications into building-block specifications in reasonable computation times. Critical design issues such as the resolution per stage are optimized in terms of power consumption and silicon area. As a case study, a 0.13μm CMOS 12bit@80MS/s pipeline ADC and a current-steering DAC for PLC front-end have been synthesized, high-level sized and analysed using the proposed toolbox.

ACKNOWLEDGMENTS

This work has been supported by the MEDEA+ (A110 MIDAS) and TIC2003-02355RAICONIF Projects.

REFERENCES

1. A. Rodríguez-Vázquez, F. Medeiro and E. Janssens (Editors): *CMOS Telecom Data Converters*. Kluwer Academic Publishers, 2003.
2. M. Gutavsson, J. J. Wikner and N. N. Tan: *CMOS Data Converters for Communications*. Kluwer Academic Publishers, 2000.
3. G. G. E. Gielen and R. A. Rutenbar: "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits". *Proceedings of the IEEE,* Vol. 88, pp. 1825-1852, December 2000.
4. K. Uyttenhove, J. Vandenbussche, E. Lauwers, G. G. E. Gielen and M. S. J. Steyaert: "Design Techniques and Implementation of an 8-bit 200-MS/s Interpolating/Averaging CMOS A/D Converter". *IEEE Journal of Solid-State Circuits*, Vol.38, pp. 483-494, March 2003.
5. K. Uyttenhove and M. S. J. Steyaert: "A 1.8-V 6-bit 1.3-GHz Flash ADC in 0.25-μm CMOS". *IEEE Journal of Solid-State Circuits*, Vol. 38, pp. 1115-1122, July 2003.
6. Y. Geerts, M. Steyaert, and W. Sansen, "A High Performance Multibit ΔΣ CMOS ADC". *IEEE Journal of Solid-State Circuits*, Vol. 35, pp. 1829-1840, December 2000.
7. A. Di Giandomenico, S. Paton, A. Wiesbauer, L. Hernández, T. Pöster and L. Dörrer: "A 15-MHz Bandwidth Sigma-Delta ADC with 11 bits of Resolution in 0.13μm CMOS". *Proc. of the 2003 European Solid-State Circuits Conference*, pp.233-236.

8. A. Shabra and H.-S. Lee: "Oversampled Pipeline A/D Converters With Mismatch Shaping". *IEEE Journal of Solid-State Circuits*, Vol. 37, pp. 566-578, May 2002.
9. W. Yang, D. Kelly, I. Mehr, M. Sayuk and L. Singer: "A 3-V 340-mW 14-b 75-Msamples/s CMOS ADC with 85-dB SFDR at Nyquist Input". *IEEE Journal of Solid-State Circuits*, Vol. 36, pp. 1931-1936, Dec. 2001.
10. E. Liu and A. Sangiovanni-Vincentelli: "Verification of Nyquist Data Converters Using Behavioral Simulation". *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, pp. 493-502, April 1995.
11. J. Goes, J. C. Vital and J. E. Franca: *Systematic Design for Optimization of Pipelined ADCs*. Kluwer, 2001.
12. P. T. F. Kwok and H. C. Leung: "Power Optimization for Pipeline Analog-to-Digital Converters". *IEEE Trans. on Circuits and Systems - Part II*, pp. 549-53, May 1999.
13. E. Bilhan, P. C. Estrada-Gutierrez, A. Y. Valero-Lopez and F. Maloberti: "Behavioral Model of Pipeline ADC by Using Simulink". *Proc. of 2001 Southwest Symposium on Mixed-Signal Design*, pp. 147-151.
14. R. Lotfi, M. Taherzadeh-Sani, M. Yaser-Azizi and O. Shoaei: "Systematic Design for Power Minimization of Pipelined Analog-to-Digital Converters". *Proc. of 2003 IEEE Int. Conf. Computer-Aided Design*, pp. 371-374.
15. The MathWorks Inc.: "MATLAB®. The Language of Technical Computing. V.6.5 R.13", July 2002.
16. The MathWorks Inc.: "Simulink®. Model-based and System-based Design. V.5 R.13", July 2002.
17. F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez: *Top-Down Design of High-Performance Modulators*, Kluwer Academic Publishers, 1999.
18. A. Van den Bosch, M. Steyaert and W. Sansen: "SFDR-Bandwidth limitations for high speed high resolution current steering CMOS D/A converters". *IEEE International Conference on Electronics, Circuits and Systems* (ICECS), vol. 3, pp. 1193-1196, 1999.
19. J. Vandenbussche, G. Gielen and M. Steyaert: *Systematic design of analog IP blocks.* Kluwer Academic Publishers, 2003.
20. J. Hyde, T. Humes, C. Diorio, M. Thomas, and M. Figueroa: "A 300-MS/s 14-bit Digital-to-Analog Converter in Logic CMOS". *IEEE Journal of Solid-State Circuits*, vol. 38, no. 5, May 2003.
21. A. Van den Bosch, M. A. F. Borremans, M. S. J. Steyaert, and W. Sansen: "A 10-bit 1-GSample/s Nyquist Current-Steering CMOS D/A Converter". *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, March 2001.
22. J. Vandenbussche, G. Van der Plas, A. Van den Bosch, W. Daems, G. Gielen, M. Steyaert, W. Sansen: "A 14 b 150 Msample/s update rate Q^2 random walk CMOS DAC". *IEEE Solid-State Circuits Conference* (ISSCC), pp. 146-147, February 1999.
23. Y. Cong and R. L. Geiger: "Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decode DAC Arrays". *IEEE Transactions on Circuits and Systems*, vol. 47, pp. 585-595, July 2000.