A Linear Programming Approach for Synthesis of Mixed-Signal Interface Elements

Adrián Núñez-Aldana and Ranga Vemuri
Electrical and Computer Engineering and Computer Science
University of Cincinnat; Cincinnati, OH 45221-0030, USA
anunez@ececs.uc.edu, ranga.vemuri@uc.edu

Abstract This paper presents a novel methodology to perform component selection and constraint transformation of analog-digital interface elements of mixed-signal systems. Sharing techniques are used to minimize area while meeting system constraints. The methodology selects components from a pre-defined library using a linear programming approach. The selection process is guided by a knowledge-based performance estimator. With the estimated parameters, linear models are generated helping the constraint transformation process and speeding up the component selection process. Experimental results show the effectiveness of the methodology in relatively short execution times.

Introduction

The synthesis of mixed-signal designs consists on three parts: (1) synthesis of digital section, (2) synthesis of analog section, and (3) synthesis of analog-digital interface section. The synthesis of digital systems is in a relative mature phase. CAD tools to synthesize digital circuits from behavioral description level have been developed for several years and relatively stable implementations of various high-level synthesis algorithms have started to emerge [10]. The synthesis of analog circuits has been relegated to hand-craft designs. Lately, some CAD tools have been developed to accelerate the analog circuit design process [1]. Tools for the design of analog components and physical layouts have been implemented in recent years. Currently, new methodologies are emerging to automatically synthesize analog systems from behavioral description level [4]. Finally, the synthesis of the analog-digital interface section has been relegated

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35498-9_57

L. M. Silveira et al. (eds.), *VLSI: Systems on a Chip*

to a separated design process; however, with the ability to integrated analog and digital circuits in a single chip, it has gained more attention.

To completely automate the design process, it is critical to embed the synthesis of analog-digital interface modules to the mixed-signal design methodologies. The interface elements establish the communication channels between the two different domains (continuous and discrete) providing means of signal conversion and synchronization. Usually the interface consists of analog-to-digital (A/D) and digital-to-analog converters (D/A), control logic, and synchronization modules. A/D and D/A converters are used to convert back and forth between the two types of signals. The control logic and synchronization modules coordinate the communication protocol between the digital and analog signals, such that the timing constraints are met.

In this paper, we present a methodology to automate the high-level synthesize of the analog-digital interface section of mixed-signal systems. The methodology is driven by the digital and analog communication requirements and system constraints. We use sharing techniques to minimize silicon area while meeting the design constraints.

1. RELATED WORK

Some effort has been done to automate the synthesis of mixed-signal systems. Most of the work has been dedicated to layout synthesis. Costa et al. [3] developed a technique to extract substrate coupling parameters in mixed-signal designs and to generate models that reflect these parameters. Miliozzi et al. [6] presented a methodology for automatic layout generation for a class of mixed-signal circuits in presence of substrate induced noise. KOAN/ANAGRAM [2] is a layout tool from CMU, which automatically generate layout for CMOS mixed-signal designs.

Different topologies of analog-to-digital and digital-to-analog converters have been published in the literature. Each topology is designed to meet different constraints. A/D and D/A converters are designed to reduce noise, speed up conversion time, be linear, consume low power, work in high range of temperatures, etc. Besides, synthesis methodologies have been developed to automate the design process of A/D and D/A converters. Sarraj [9] presents a technique to design high speed pipeline A/D converters. Johnston [5] describes a procedure to calibrate features in Delta-Sigma A/D converters. Morling et al [7] presents the design of a sigma delta CODEC for telecommunication applications. However, A/D and D/A converters have not been integrated to the synthesis of mixed-signal designs.

2. SYNTHESIS OF A-D INTERFACE SYSTEMS

The analog-digital interface elements are constrained by the analog and digital communication requirements. The sampling theorem states that, to avoid aliasing, the sampling frequency f_s must be higher that two times the bandwidth of the input signal. Therefore, the conversion time should be less than $T_s = 1/f_s$. Besides, specifications of acceptable quantization error determine the number of bits required for the converters. Topologies and transistor sizes determine the area and performance of the interface elements. The selection process should select interface elements from a component library such that those and other constraints are met while minimizing silicon area.

Digital-to-analog converters. Several topologies of D/A converters have been published in the literature. Each topology is characterized by static and dynamic properties. Static properties establish linearity, resolution, zero and full-scale error, and monotonicity of the converters. Dynamic properties describe the behavior of the converters when the input word is changing. The settling time is a dynamic property which determines the response time.

Table 1 shows characteristics of four different D/A converters topologies. Style 1 and style 2 are voltage-scaling approaches. They have the advantages of being fast and insensitive to parasitic capacitances, but they are non-linear and require large silicon area. Style 3 is a charge-scaling converter. As it is based in switched-capacitor technology, it is very accurate, but it is limited to medium speed applications. Style 4 uses an algorithmic approach. It uses small area but it is very slow.

Analog-to-digital converters. Table 2 shows four different A/D converter topologies. Style 1 is a serial converter, therefore it is the slowest, but it uses small silicon area. Using a successive approximation approach, style 2 speeds up the conversion time of style 1. The fastest A/D converter is the style 3. It makes the conversion in parallel reducing the conversion time to one operational amplifier propagation delay. However, it has the disadvantage to use large silicon area. Style 4 uses a $\Sigma\Delta$ modulator to perform the conversion. It has the ability to be easily integrated on a chip that is predominantly made up of complex digital circuitry.

Estimation and model geneartion of A/D - D/A. The area and speed of the converters depend on the topology, number of bits and sizing of the analog components. To determine the performance of each converter, the components should be sized first. Then, a wide range of converters with different performance parameters can be generated based on the topologies presented in previous sections by changing the circuit sizing. However, circuit sizing process is computational expensive, and the possible combinations increase

Table 1 Characteristics of digital to analog converters

	Style 1	*Style 2*	*Style 3*	*Style 4*
Name	Weighted R	R-2R	Weighted C	Serial
Adv.	Fast Insensitive	Fast Small elements	Best Accuracy	Min area
Disadv.	Large elements Precision R's	Precision R's Non-monotonic	Large elements Non-monotonic	Slow Complex
Area (A)	$\sum_{i=0}^{N-1} 2^i A(R)$ $+\frac{1}{2}A(R)$ $+N * A(Sw)$ $+A(OA)$	$3N * A(R)$ $+\frac{1}{2}A(R)$ $+N * A(Sw)$ $+A(OA)$	$\sum_{i=0}^{N-1} A(\frac{C}{2^i})$ $+A(\frac{C}{2^{N-1}})$ $+(N+2)A(Sw)$ $+A(OA)$	$2 * A(C)$ $+4 * A(Sw)$ $+A(OA)$ $+A(logic)$
Power (P)	$V_{ref}^2 \sum_{i=0}^{N-1} \frac{1}{2^i R}$ $+P(OA)$	$\frac{Vref^2}{R}$ $+P(OA)$	$P(OA)$	$P(OA)$
Speed	$Delay(Sw)$ $+\Delta T(OA)$	$Delay(Sw)$ $+\Delta T(OA)$	$Delay(Sw)$ $+4.6RC$ $+\Delta T(OA)$	$2NDelay(Sw)$ $+4.6NRC$ $+\Delta T(OA)$

Note: N specifies the number of bits of the converters

exponentially as thea number of possible sizing elements increase. Therefore, critical to the success of our methodology is the accuracy and speed of an analog performance estimator. We use an analog performance estimator (APE) [8] to evaluate the performance of the interface modules.

To automate and speed up the design process, we generate linear performance models of the constituent components of the A/D and D/A converters presented in previous sections. Sweeping a wide range of the design parameters, we estimate the performance of the components using APE. Then, using an interpolation method, we generate linear component performance models of the following format:

$$perf = \alpha_1 * designP_0 + \alpha_2 * designP_2 + \ldots + \alpha_r * designP_r + \alpha_{r+1} \quad (1.1)$$

Formulation of the problem. We assume that the number of analog-to-digital and digital-to-analog conversions, sampling frequency (f_s) and number of bits of the converters (N) are defined by the user. Let's start by formulating the problem for the A/D converters first, then the D/A converters.

Table 2 Characteristics of analog to digital converters

	Style 1	*Style 2*	*Style 3*	*Style 4*
Name	Serial	Succ. Approx.	Flash	Over Sampling
Notes	Low Complex	Low Complex	Fast	Complicate
	High resolution	Medium speed	Large area	Medium speed
	Slow	Medium area	Low resolution	high SNR
Elem	Ramp Generator	N-bit DAC	$(2^N - 1)$ opamps	Modulator
	2 N-bit Count's	opamp	2^N resistors	Decimator
	opamp	N-bit Register	$(2^N - 1)$ decoder	LP Filter
		N-bit Shift Reg.		
Area	*(Small)*	*(Medium)*	*(Large)*	*(Small)*
(A)	$A(ramp)$	$A(DAC[N])$	$\sum_{i=1}^{2^N-1} A(opamp)$	$A(Mod)$
	$+2A(count[N])$	$+A(opamp)$	$+2^N A(R)$	$+A(FF)$
	$+A(opamp)$	$+A(Reg[N])$	$+A(dec[2^N - 1])$	$+A(LPF)$
		$+A(Shift[N])$		
Speed	$2^N * \mathrm{T}$	$N * \mathrm{T}$	$1 * \mathrm{T}$	$4 * N * \mathrm{T}$
Clk	$Delay(DAC)$	$Delay(DAC)$	$\Delta T(opamp)$	$Delay(\Sigma\Delta)$
	$+\Delta T(opamp)$	$+\Delta T(opamp)$	$+Delay(dec)$	$+Delay(FF)$
	$+Delay(logic)$	$+Delay(logic)$		

Analog-to-digital converters. Let's define S to be the set containing all analog signals which are read by the digital section in a mixed-signal system. Then, $p = ||S||$ is the number of signals that require an A-to-D conversion.

Several frequencies are used in mixed-signal systems. First, the sampling frequency $f_s = \frac{1}{T_s}$ is the rate of conversion defined by the sampling theorem. T_s constrains the conversion time CT between analog and digital signals, $CT < T_s$. Second, Let $f_d = \frac{1}{T_d}$ be the clock frequency of the digital logic. The digital logic requires to read all analog-to-digital ports at the sampling frequency; therefore, $p * T_d < T_s$. Finally, the A/D converters operating frequency $f_c = \frac{1}{T_c}$ determines the speed ratio of the converters. T_c is a function of the A/D converter topology, sizing, and fabrication process parameters.

The propagation delay T_i of the i^{th} converter can be estimated as follows:

$$\begin{aligned} T_{c_i} &= \textstyle\sum_{k \in ADC_i} DelayOf(k) \\ &= \textstyle\sum_{k \in ADC_i} (\alpha_{1_k} * designP_0 + \ldots \alpha_{r_k} * designP_r + \alpha_{r+1_k}) \qquad (1.2) \\ &= \textstyle\sum_{j=1}^{n} (a_{ij} * P_{ij}) + b_i \end{aligned}$$

Where P_{ij} is the low-level design parameter j of the converter type i, and a_{ij} and b_i are constants defined as follows:

$$\begin{array}{l} a_{ij} = \sum_{k \in ADC_i} \alpha_{j_k} \\ b_i = \sum_{k \in ADC_i} \alpha_{r+1_k} \end{array} \quad (1.3)$$

Similarly, we can estimate the $Area_i$ of the i^{th} converter (with c_{ij} and d_i being the constants introduced by the performance models) as follows:

$$Area_i = \sum_{j=1}^{n} (c_{ij} * P_{ij}) + d_i \quad (1.4)$$

Let's define C to be a function which maps the A/D converter type and the number of bits N to number of cycles T_c required to perform the conversion: $C : Type \times NoBits \rightarrow NoOfCycles$:

$$C(i,N) = \begin{cases} 2^N & \text{for i=1} \quad \text{(Serial)} \\ N & \text{for i=2} \quad \text{(Successive)} \\ 1 & \text{for i=3} \quad \text{(Parallel)} \\ 4 * N & \text{for i=4} \quad (\Sigma\Delta) \end{cases} \quad (1.5)$$

CT is a function from the converter type to the conversion time:

$$CT(i,N) = C(i,N) * T_{c_i} \quad (1.6)$$

We define the maximum load of a converter as follows:

$$Load_{i \times N} = \left\lfloor \frac{T_s}{CT(i,N)} \right\rfloor \quad (1.7)$$

Finally, we model the synthesis of mixed-signal interface elements as an optimization problem. The component selection process selects elements from the analog component library such that the area is minimized while all the constraints are met. Let X_i be the number of converters type i selected (X_i is defined to be a positive integer number). Assuming the library contains u A/D converters, the optimization and constraint functions are as follows:

$$\begin{array}{ll} minimize & Area = \sum_{i=1}^{u} Area_i * X_i \\ ST & \sum_{i=1}^{u} Load_{i \times N} * X_i \geq p \end{array} \quad (1.8)$$

Digital-to-analog converters. Similarly, we formulate the problem of selecting D/A converters. Let's define Q to be the set containing all digital output signals communicating with the analog section, and $q = ||Q||$ the number of digital signals which require a digital-to-analog conversion. The digital logic requires to write to all digital-to-analog ports at the sampling frequency; then, $q * T_d < T_s$. The conversion time T_c, area $Area_i$ and load $Load_{i\times N}$ of the D/A converters are defined similarly to the A/D converters. Assuming the library contains v D/A converters, the optimization and constraint functions are as follows:

$$\begin{array}{ll} minimize & Area = \sum_{i=1}^{v} Area_i * X_i \\ ST & \sum_{i=1}^{v} Load_{i\times N} * X_i \geq q \end{array} \quad (1.9)$$

Methodology. To solve an integer linear problem, the object function, constraint and right-hand coefficients should be known. However, in equations (1.8) and (1.9), $Area_i$ and $Load_i$ are functions of the topology, sizing and process parameters. Evaluating all solutions are infeasible; therefore, we need to determine how changes in the parametric values affect the optimal solution, such that only parametric ranges are evaluated. Sensitivity analysis determines ranges for the coefficients of a linear model such that the set of basic varibles in the optimal solution is unchanged, although their values may change.

3. EXPERIMENTAL RESULTS

In this section, we present experimental results involving circuit synthesis of mixed-signal interface elements using the methodology presented in this paper. Table 1.3 shows the synthesis results of several real life examples. Circuit 1 is a T1 channel service unit with 24 full duplex voice channels. Circuit 2 calculates the real, RMS and reactive power of a three phases electrical line. Circuit 3 is a bank of 16 digital filters. Circuit 4 is a programmable logic controller which has 8 analog outputs and 4 analog inputs. Circuit 5 is an example of a high-speed communication channel which uses the asynchronous transfer mode. Finally, circuit 6 is an equation solver.

Table 3 shows the execution time required to generate the models, and perform the selection process. The model generation is a one-time process. The last column shows the number of converters selected for each example. In all cases, the total number of converters is less than the total number of A/D and D/A signals, which indicates that converters are being shared.

4. CONCLUSIONS

We present a methodology to perform the component selection and constraint transformation process for analog-digital interface elements of mixed-signal systems. The problem was formulated as an integer linear programming model. Using sensitivity analysis, we speed up the selection process avoiding the evaluation at every single solution point. The methodology is guided by an analog performance estimator, which is used to generate performance models. In future work, we plan to extend the methodology to work with non-linear

Table 3 Results of synthesis of mixed-signal interface elements

ckt	D/A conv	A/D conv	T_s	model gen	exec time	area	no. conv.
			$(\mu sec.)$	$(sec.)$	$(sec.)$	(μ^2)	
T1	24	24	125	9.62	40.01	51200	12
Power	3	6	4000	9.62	28.87	10988	2
Filter	16	16	5	9.62	37.13	151848	15
PLC	8	4	100	9.62	29.39	24188	5
ATM	4	4	2	9.62	35.02	101825	6
Eq	32	32	10	9.62	33.89	99125	18

models, and use interleaving techniques to meet fast conversion time application requirements.

5. REFERENCES

[1] L. R. Carley, G. Gielen, R. A. Rutenbar, and W. Sansen. Synthesis tools for mixed-signal ics: Progress on frontend and backend strategies. *proceedings of Design Automation Conference*, pages 298–303, June 1996.

[2] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley. Koan/anagram ii: New tools for device-level analog placement and routing. *IEEE JSSC*, 26(3), March 1991.

[3] J. P. Costa, M. Chou, and L. M. Silveira. Efficient techniques for accurate extraction and modeling of substrate coupling in mixed-signal ic's. *Conference proceedings of Design Automation and Test in Europe*, pages 396–400, March 1999.

[4] A. Doboli, A. Nunez-Aldana, N. Dhanwada, S. Ganesan, and R. Vemuri. Behavioral synthesis of analog systems using two-layered design space exploration. *36th Design Automation Conference*, June 1999.

[5] J. E. Johnston. Calibration features in delta-sigma a/d converters. *Proceedings of Analog & Mixed-Signal Applications Conference*, pages 209–213, July 1998.

[6] P. Miliozzi, I. Vassiliou, E. Charbon, and E. Malavasi. Use of sensitivities and generalized substrate models in mixed-signal ic design. *Proceedings of Design Automation Conference*, pages 227–232, June 1996.

[7] R. C. Morling, I. Kale, C. W. Tsang, S. J. Morris, G. Hague, and C. Foran. The design of a sigma delta codec for mobil telephone applications. *Second International Conference onf Advanced A-D and D-A conversion techiques and their applications*, July 1994.

[8] A. Nunez-Aldana and R. Vemuri. An analog performance estimator for improving the effectiveness of cmos analog systems circuit synthesis. *Proceedings of Design, Automation and Test in Europe*, pages 406–411, March 1999.

[9] M. Sarraj. New high speed technique for pipeline adc design. *Proceedings of Analog & Mixed-Signal Applications Conference*, pages 205–208, July 1998.

[10] R. Walker and R. Camposano. A survey of high level synthesis systems. *Kluwer Academic Publishers*, 1991.