High-level synthesis of analog sensor interface front-ends

S. Donnay*, G. Gielen†, W. Sansen
Katholieke Universiteit Leuven
Dep. Elektrotechniek, ESAT-MICAS
Kardinaal Mercierlaan 94, B-3001 Heverlee
Belgium

W. Kruiskamp, D. Leenaerts, W. van Bokhoven
Eindhoven University of Technology
Dept. of Electrical Engineering
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

Abstract

In this paper we will compare three different methodologies for analog high-level synthesis. Two optimization-based methods—one with simulations in the loop, the other with equations—and a library-based approach are discussed and illustrated with experimental results. The comparison is made by means of a real life design example—a radiation detector interface ASIC—although the methodologies presented in this paper, are generally applicable.

1 Introduction

Advances in VLSI technology allow the integration of very complex signal processing systems on a single ASIC. In many cases these systems in silicon contain analog and digital signal processing circuits as well as digital controllers on the same die. An important application domain for these mixed analog/digital integrated systems are sensor interfaces. Although there is a strong trend towards implementing as much signal processing functionality as possible in the digital domain, analog circuits will always play an important role in these interfaces, e.g. for amplification and filtering of the noise-sensitive sensor signals and for A/D conversion. A very important problem in the design of these mixed-signal sensor interfaces is the lack of mature analog CAD tools, especially at levels higher than the opamp level, resulting in unacceptably long design times for the analog sensor interface front-end.

In [1] a methodology was discussed for the design automation of the analog part in mixed-signal ASICs. The high-level synthesis of the analog part was identified as a very important problem in a complete mixed-signal design environment. In this paper we will discuss and compare three different approaches for the analog high-level synthesis problem. The comparison will be made by means of a real life design example—a radiation detector interface—although the methodologies that will be presented are generally applicable.

*since January 1997 with IMEC, Kapeldreef 75, B-3001 Leuven
†research associate of the National Fund of Scientific Research

In section 2 the design example will be briefly introduced. Next, in section 3, the analog high-level synthesis problem will be discussed and the three different approaches will be introduced: two optimization-based approaches (the simulation-based and the equation-based approach) and the library approach . Finally, in section 4 conclusions are drawn.

2 Design case: radiation detector interface

Radiation detectors are used in nuclear physics experiments or for on-satellite radiation measurements [2]. The purpose is to measure the energy of particles that hit one or several detectors. The radiation detectors are sensors that generate small charge pulses that are proportional to the energy of the incoming particles. The radiation detector interface front-end provides the link between the sensors and a digital signal processing core. Each time a particle hits one of the detectors, charge packets are generated and have to be collected and transformed into a voltage that is equivalent to the total charge generated in the detector. When that voltage exceeds a user-programmable threshold, the ”event” has to be recorded: the voltage is converted into a digital word which can further be processed by the DSP core.

Figure 1 shows a general architecture of a radiation detector interface [3]. The sensor signals are very weak and noise-sensitive. In a first step the sensor signals are amplified and filtered in an analog preprocessing chain. This analog chain consists of a charge sensitive amplifier (CSA) and a pulse shaping amplifier (PSA). The charge packets coming from the detectors are amplified and integrated on a capacitor in the CSA and then shaped to a semi-Gaussian pulse in the PSA. The PSA is a bandpass filter consisting of a differentiator and a number of n integrators.

The output of the analog chain (CSA-PSA) is fed to a peak detect sample and hold (PDSH) circuit which holds the peak value of the pulse until it is converted by an analog-to-digital converter (ADC). In this way the PDSH functions as an analog memory as indicated in figure 1. The output of each analog chain is also fed to an event trigger. This

ED&TC ’97 on CD-ROM

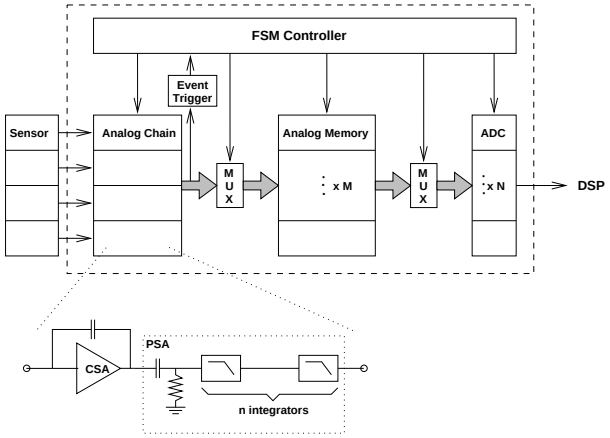

Figure 1: General radiation detector interface architecture.

is a comparator which compares the signal with a user-programmable threshold . When the peak value of the pulse exceeds this threshold level, a digital pulse is sent to the digital controller indicating that a detector has been hit. The controller is implemented as a finite state machine (FSM), which controls the signal flow in the interface architecture.

3 Analog high-level synthesis

Analog high-level synthesis is the translation of analog system-level specifications into an architecture, in which all the sub-blocks have been completely specified. For the high-level synthesis of analog architectures a methodology was introduced in [1]. The methodology, shown in figure 2 consists of three main steps: architecture selection/generation, specification translation and behavioral verification. The flow of the radiation detector interface design with the high-level synthesis tool is schematically shown in figure 2. First the user has to specify the functionality and performance specifications of the system to be designed. Then the high-level synthesis tool selects the optimal architecture and translates the system specifications into optimal specifications of the sub-blocks.

After the high-level design step, the resulting architecture has to be verified by means of behavioral simulations. Next, the different modules can be synthesized separately. The finite state machine controller can be synthesized with a commercial logic synthesis tool starting from a VHDL description. The analog modules can be generated by means of an analog module generator (AMG), like the one described in [4]. The output of these tools is a complete transistor-level design and layout. In a bottom-up layout assembly and detailed verification step the complete chip layout can be finalized. If one of the verifications in figure 2 shows that the system specifications are not met, a redesign has to be initiated by choosing another architecture

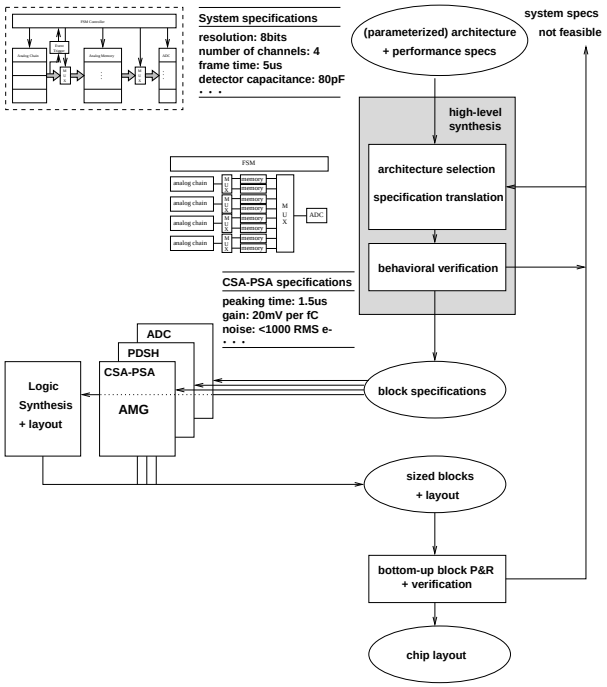

Figure 2: Design of the radiation detector interface ASIC with the high-level synthesis tool.

or repartitioning the module specifications. If no solutions can be found after a number of redesigns, the system specifications are probably not feasible and the user can start the high-level synthesis again with a different set of specifications.

This generally applicable methodology will be illustrated with the design of a particular sensor interface in this paper. In the design example of section 2 a number of high-level architectural design decisions have to be made, although the generic block diagram is already known (see figure 1). The main architectural parameters are: the number of ADCs (N), the number of memory cells per channel (M) and the order of the PSAs (n). Next to the determination of a proper architecture, the sub-block specifications have to be determined in such a way that all system-level specifications are met and that the power (or area) consumption of the complete system is minimal.

The most important system-level specifications for the radiation detector interface are speed, accuracy, sensitivity and power consumption. The speed is specified in terms of the processing time of an analog chain (frame time). Particles that hit the same detector in a time interval smaller than the frame time, are measured as a single particle with an energy equal to the sum of the energy of all the particles. If the analog chains are faster, then the probability of occurrence of these type of errors is reduced (at the cost of more power consumption). The accuracy and sensitiv-

ity requirements are determined by the energy range and resolution of the detectors. Typically, in nuclear physics experiments a (very) large number of detectors is processed in parallel. Therefore, the power (and area) consumption of a single channel have to be minimized as much as possible. The same is true for satellite payloads. On the other hand, speed and accuracy can only be increased at the cost of more power. This is an important high-level trade-off the designer has to make.

We will now discuss three different approaches for the high-level synthesis of the radiation detector interface. All three approaches can be mapped onto the general methodology of figure 2.

3.1 Optimization-based approaches

Both the "simulation-based" and the "equation-based" approaches discussed further, make use of a global optimization routine. In these approaches a (behavioral) performance evaluation of the analog system under design is used inside the loop of a simulated annealing optimization algorithm, as shown schematically in figure 3.

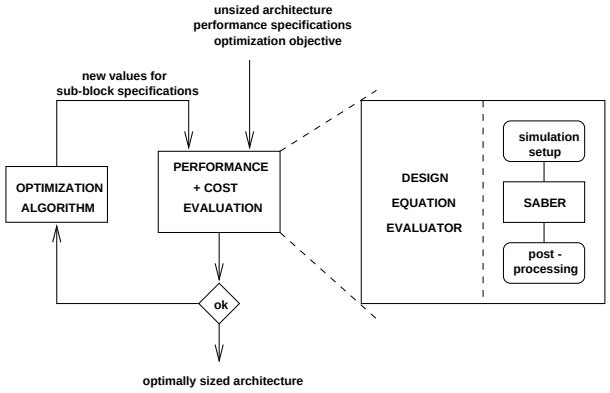

Figure 3: Simulated annealing-based optimization loop.

The optimization variables are the design parameters of the sub-blocks used in the analog architecture, e.g. the number of bits and speed of the ADCs or the gain, speed and noise performance of the CSA and PSA. The cost function being minimized during optimization contains two sorts of terms: (1) terms related to the difference between the system performance specifications and the evaluated system performance values (for a particular set of sub-block parameters) and (2) terms related to the general objectives that have to be minimized at the same time, such as power or area. For the different sub-blocks used in the radiation detector interface, heuristic equations have been developed that give an estimate of the power and area consumption as a function of the performance specifications of these sub-blocks. Equation (1) for example, gives a formula for the power of an ADC [5]:

$$P_{adc} = FOM \cdot DR \cdot SR \cdot 10^{12} \qquad (1)$$

where *DR* is the ADC's dynamic range and *SR* is the sampling rate (in MHz). *FOM* is the figure of merit of the ADC. In [5] the *FOM* of several recently published ADCs are listed.

3.1.1 Simulation-based approach

In this approach the performance evaluation inside the optimization loop of figure 3 is implemented by means of behavioral simulations, e.g. with SABER [6] or ELDO [7]. Behavioral models were developed for all sub-blocks used in the radiation detector interface. The sub-block performance specifications are the behavioral model parameters. The user of the tool has to describe the architecture as an interconnection of these sub-block models, define which sub-block parameters have to be varied during optimization and specify the bounds for each of these parameters. He also has to provide the system performance specifications and a simulation plan for each of these system performances. Such a simulation plan includes the test setup (input sources, loads, feedback, etc.), the input signals and simulation commands and the required data processing on the simulation results to obtain the wanted system performance.

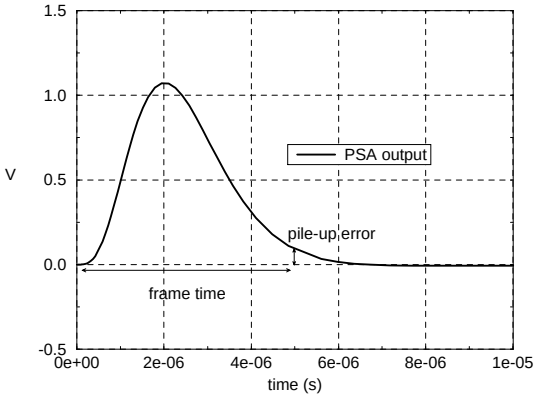

Figure 4: Pile-up error introduced by CSA-PSA.

As an example figure 4 shows the simulation of one of the important sources or errors in the detector interface system, the pile-up error at the output of the PSA. It is a transient simulation of the analog preprocessing chain, where the pile-up error is found by measuring the residual signal at the output of the PSA at the end of the frame time.

A very important problem in this approach is the required simulation time of one performance evaluation. Since simulated annealing typically takes several tens of

thousands of iterations, the execution time of the global optimization will be unacceptable, when one performance evaluation (consisting probably of several simulations) takes more than a few seconds. An important advantage of the approach is that the user can enter a new high-level synthesis problem into the system in a minimum amount of time (if all sub-block behavioral models are available).

3.1.2 Equation-based approach

A solution to the execution time problem is to replace the lengthy simulations with design equations. These equations directly express the performance of the system as a function of the sub-block specifications. As an example we give the equation that can replace the simulation of figure 4. The output voltage of the PSA at $t = T_{frame}$ for an ideal step input at $t = 0$ is given by:

$$V_{pile-up} = \frac{Q \cdot A_{csa} \cdot A_{psa}^{n} \cdot n^{n}}{n!} \left(\frac{T_{frame}}{\tau_s} \right)^{n} e^{-n \cdot T_{frame}/\tau_s} \tag{2}$$

in which Q is the amount of charge generated by the infalling particle, A_{csa} is the gain of the CSA, A_{psa} is the gain of one PSA stage, n is the order of the PSA and τ_s is the peaking time.

An important advantage of this approach is that the evaluation of a set of equations is a lot faster than the execution of the simulation plan of the previous approach. This can reduce the execution time of a complete optimization from several hours to only a few minutes. The disadvantage is the much larger setup time. The user has to derive all necessary design equations, which can be a very difficult and time consuming as well as error-prone task. Sometimes the behavior that has to be captured in design equations may be analytically so complex, that approximations have to be used. Furthermore, the design equations are often very specific for the system architecture and cannot be reused for other architectures. In the simulation-based approach on the other hand, the behavioral modeling effort is situated at the sub-block level instead of on the system level, leading to a wide reusability of the sub-block models in different architectures.

3.1.3 Architectural optimization

In the optimization-based approaches the user can either describe the architecture completely or as a parameterized architecture—parameters which will also be varied during the optimization, e.g. the number of ADCs, number of memories, etc. These architectural parameters will also be changed during optimization, in order to find an optimal architecture at the same time as optimal sub-block specifications.

Figure 5 shows the results of a number of optimizations with different speed and accuracy requirements for the radiation detector interface. The speed requirement is represented in kHz (inverse of the frame time) and the accuracy requirement is plotted as the dynamic range (in dB) of the system. Figure 5 also shows which architectures have been selected by the high-level synthesis tool (because of power minimization) for the different combinations of specifications.

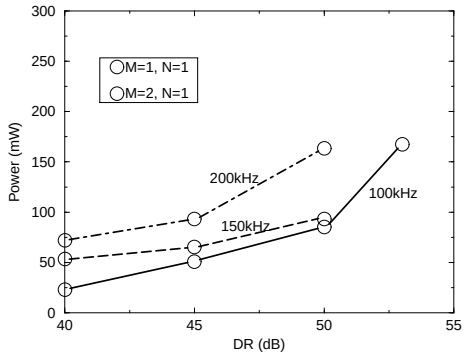

Figure 5: Optimal architectures for different system requirements.

3.2 Library approach

An alternative is the template approach. In this approach a number of fixed architectures, together spanning a wide range of system specifications for a specific application domain, are stored as “templates” in a library, together with design plans containing all necessary design knowledge required for architecture selection and high-level specification translation. In this way the tremendous modeling effort is taken out of the hands of the user of the tool; the derivation and coding of all the design knowledge has been performed by the tool developer. This is an interesting approach when it is anticipated that the architectures will be used in a large number of designs with different specifications, hence justifying the large initial development effort. Also the selection of an appropriate architecture from the library can now be automated. This allows user interaction at a much higher level of abstraction. All that the user needs to supply are the system specifications. Therefore, such a tool can also be used by non-experienced users.

The design knowledge stored in the library templates is of course essential in this approach. A template contains three types of information [8]: (1) *architecture knowledge*, which is organized hierarchically. (2) *specification translation knowledge* in the form of equations expressing the system-level specifications as function of the sub-block specifications. (3) *performance knowledge* of all architectures and sub-blocks. Regions have been derived for each block, called “feasibility spaces”, containing all fea-

sible combinations of block specifications. Most of the block specifications are mutually dependent, resulting in very complex descriptions for these feasibility spaces.

Specification	Unit	Value
Radiation detector capacitance	pF	80
Number of input channels	-	4
Technology	-	0.7μm CMOS
Positive power supply	V	2.5
Negative power supply	V	-2.5
Maximum energy	keV	2500
Minimum energy	keV	10
Average hit rate per detector	$1/ms$	200
Maximum error-rate	%	5
Maximum area	mm^2	25
Maximum power consumption	mW	1000

Table 1: System-level specifications for the radiation detector interface ASIC.

Specification	Unit	Value
Analog processing chain:		
Processing time	μs	0.265
SNR	dB	60
Maximum area	mm^2	1
Maximum power consumption	mW	125
ADC:		
Conversion time	μs	1
Number of bits	-	8
Maximum area	mm^2	4
Maximum power consumption	mW	125

Table 2: Sub-block performance specifications.

An important drawback of the template approach is the enormous work which must be done in order to store the templates with all the design knowledge into the libraries. The main advantage on the other hand, is the computational speed, which allows the user to make many design iterations to investigate high-level trade-offs. The design of a radiation detector interface for the specifications of table 1 was completed within a few seconds. The tool selected an architecture with 2 ADCs and 1 memory cell per channel. The resulting specifications for the ADCs and the analog processing chains are given in table 2. In the next step, the same procedure was repeated to design the analog chain: selection of an architecture and computation of the specifications for CSA, PSA, etc.

4 Conclusion

In this paper three different approaches towards analog high-level synthesis have been discussed and compared to each other, by means of a real life design example. Some important conclusions are summarized in table 3.

The template approach is the best choice when it is anticipated that a large number of similar designs have to be made in a certain application domain. In that case the large initial effort of library setup can be written off over a number of designs.

	Simulation-based	Equation-based	Library templates
execution time of one high-level synthesis run	hours	minutes	seconds
setup time of new design	hours–days (by user)	weeks (by user)	months (by tool developer)
required user design experience	high	high	low
reusability of modeling in other architectures or applications	high	low	low

Table 3: Comparison of the three approaches for analog high-level synthesis.

When an open system is required, where the designer can tackle new design problems quickly, the optimization-based approaches are more promising. In that case, when we take both setup time and execution time into account, we can conclude that a combined approach—simulations as well as equations in the performance evaluation—is the most optimal solution in most cases. Simulations can be used when they do not consume too much CPU time (or when no equations can be found), and equations can be used to replace lengthy simulations and/or when their derivation is not too difficult.

Acknowledgments

This research was supported by ESA-ESTEC.

References

[1] S. Donnay, K. Swings, G. Gielen, W. Sansen, W. Kruiskamp, D. Leenaerts, “A methodology for analog design automation in mixed-signal ASICs”, *Proc. ED&TC*, 1994, pp. 530–534.

[2] WIND-SST Project, ESA-ESTEC, Noordwijk, The Netherlands.

[3] Z. Chang, W. Sansen, *Low-noise wide-band amplifiers in bipolar and CMOS technologies*, Kluwer Academic Publishers, 1991.

[4] G. Gielen et.al., “An analog module generator for mixed analog/digital ASIC design”, *International Journal of Circuit Theory and Applications*, pp.269–283, July-August 1995.

[5] F. Goodenough, “ADCs move to cut power dissipation”, *Electronic Design*, January 9, 1995.

[6] Analogy Inc., *Saber 4.0 User Manuals*.

[7] Anacad Inc., *ELDO User Manuals*.

[8] W. Kruiskamp, *Analog design automation using genetic algorithms and polytopes*, Ph.D. thesis, Eindhoven University of Technology, 1996.