ANALOG/DIGITAL CO-DESIGN

Frank Heuschen
Klaus Waldschmidt

Analog/Digital Co-Design is the central part of a design methodology for mixed signal systems. Additional degrees of freedom, provided by an enlarged design space, are mapped onto a space of possible implementations. An algorithmic analysis of this implementation space is used to find an optimized choice of architecture and an optimized partitioning.
This paper presents the basic design flow of Analog/Digital Co-Design, a graph based model of implementation space and the application of decision theory and optimization algorithms to this model.

1. Introduction

Computer aided design of embedded systems is becoming more and more important for economic and scientific reasons. Because of the accelerated progress in digital technology, approaches in design automation are mainly focused on the digital core and its software. This has led to sophisticated solutions for Hardware/Software Co-Design [4,6].

Most embedded systems reside however in a continuous time and value environment, yet the aspects of analog signal preprocessing and signal conversion are usually neglected. Pre-configured converters are inserted at the system boundaries, often as an afterthought.

Methodical approaches to mixed signal system design are few and existing design tools provide hardly automated design flows. The combined treatment of analog and digital architectures is very promising, because it opens up a considerable optimization potential. On the other hand the combined view presents an enlarged design space that is usually hard to explore at the designer's own discretion. The main objective of design space exploration is to find a configuration that satisfies functional requirements without violations of nonfunctional constraints and with efficient usage of given resources. Commonly used iterative strategies lead the designer eventually to a suitable solution, but iterative refinement is time consuming and a huge amount of optimization potential is probably wasted.

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: 10.1007/978-0-387-35409-5_23

B. Kleinjohann (ed.), *Architecture and Design of Distributed Embedded Systems*

Analog/Digital Co-Design as a methodology is an analogy to Hardware/Software Co-Design as well as an extension. We present one possible scenario of an automated Analog/Digital Co-Design process, a design flow for mixed signal systems with an emphasis on signal processing embedded systems. The central idea is to transform a given design space into a graph modeled implementation space. Implementation space can be defined as design space annotated with cost values. These values are derived from nonfunctional properties of different implementation possibilities. Any criterion that can be expressed with linear cost functions (chip area, power consumption or simply price) may be combined to a multi objective decision set.

Cost values are obtained either by special estimation methods or from the manufacturer's cell libraries. The graph model allows us to analyze the implementation space with optimization algorithms leading us to a variety of valid solutions. Because architectures and partitions are not iteratively chosen, checking of constraint violation after each decision can be avoided.

This paper presents a general description of the design flow in Analog/Digital Co-Design with an emphasis on the above mentioned implementation space exploration. The graph model that enables algorithmic optimization is introduced as embedded system graph.

The application of simulated annealing is described as well, as appropriate multi objective decision methods. An example concludes the paper.

2. Design Flow

In this section, the design flow of analog/digital co-design is presented in more detail.

On system level, the essential task of an embedded system design methodology is the mapping of the specification on the one hand onto the given design platform on the other hand.

The specification contains behavioral descriptions of signal processing functionalities and/or algorithms. The design platform contains a set of implementable subsystems. These are filed either as *IP-components*, as *system level models* or as *parametrizable templates*.

The first step is to analyze the specification and to identify functionally independent susbsystems. These subsystems are assigned with all library objects, that can perform that given functionality. These are the so called implementation possibilities. This way, we reduce the design platform to a given design space for the specified system.

To hold the design space and to perform the appropriate mapping process, denoted by the swirl in figure 1, we propose a graph based system model, that models the specified system and the directed signal flow with a two dimensional graph. The graph is then expanded with a third dimension to contain the *space* of

possible implementations. The model is called *systemgraph* and will be explained in the next chapter in more detail.

The second step, again symbolized by the swirl in figure 1, is the assessment of design space with nonfunctional properties or costs. The specification is processed now in more closely and functional demands are transformed to nonfunctional properties of the contained implementation possibilities. The values can be extracted from IP datasets or, which is our approach, the values can be calculated by specific estimation functions, that are based on the use of parametrizable templates. The values are annotated to the implementation possibilities and so the systemgraph is a model for the implementation space also.

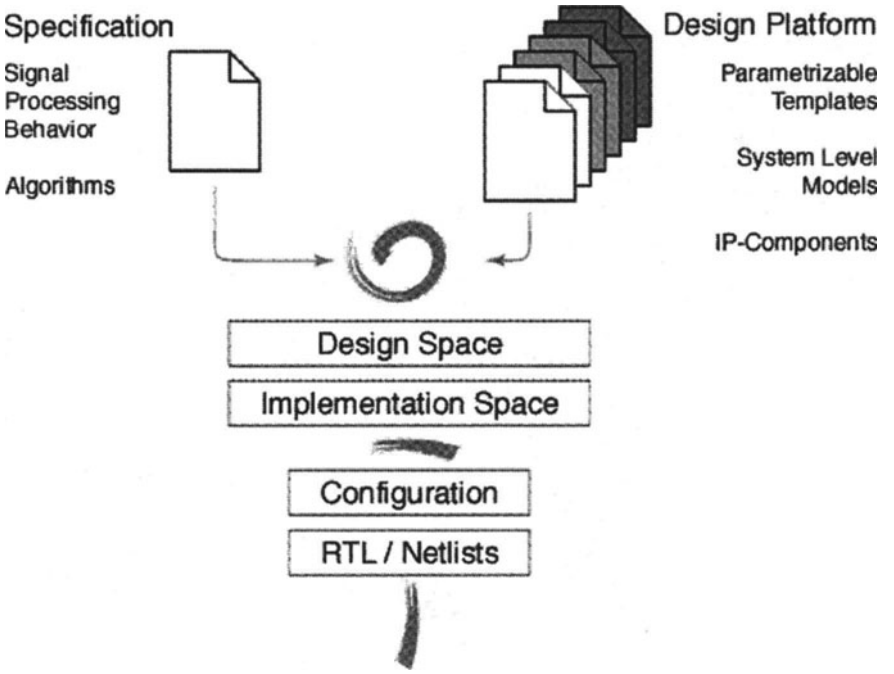

Figure 1. Design Flow

After the complete build of implementation space, the analyzing step is performed, that produces the configuration, that is worth an implementation. The analyzing Step is a combination of combinatorial optimization and application of decision theory in case we look at more than one nonfunctional property, Which is probably always the case with electronic designs.

Subsystems are represented by now with either RTL-descriptions for digital parts or net-lists for analog parts and converters. The subsequent synthesis steps are common practice and lie not within our research interest.

3. Implementation Space

3.1. Graph Model (informal description)

The implementation space of a mixed signal system is modeled by a graph, consisting (as usual) of nodes and edges. This graph is named embedded systemgraph as mentioned above. Its properties are now explained in an informal way. For the formal definition please refer to [10] or [9].

Five different signal classes occur in a mixed signal system as shown in Table 1.

Table 1. Signal classes in a mixed signal system

Signal	*Time*	*Value*	*Bandwidth*
A	continuous	continuous	unlimited
B	continuous	continuous	limited
C	discrete	continuous	limited
D	discrete	discrete	limited
G	continuous	discrete	limited

A signal processing system can be structured into functionally independent subsystems. This is done on an abstract behavioral level of specification, regardless of the processed signal class. These functionally independent subsystems are modeled by the graph's nodes.

In addition, each closed path of directional signal flow has to start and to terminate with an environmental node. These nodes carry no functionality (NOP) but they produce an input signal and its specifications (source) or they consume and define an output signal of the system (sink). The system's signal flow is now depicted as a kind of block diagram (See Fig.5).

Table 2. Possible functional nodes

Symbol	*Function*
NOP	No operation, used for environmental nodes
LP	Lowpass
HP	Highpass
BP	Bandpass
BS	Bandstop
INT	Integration
INTDT	Integration, forced analog
INDTN	Integration, forced digital
DIF	Differentiation
DIFDT	Differentiation, forced analog
DIFDN	Differentiation, forced digital
ADD	Addition
SUB	Subtraction
MUL	Multiplication
DIV	Division
ALGO	Algorithmic specified subsystem

In the current stage of implementation functionalities are identified as listed in (the expandable) Table 2.

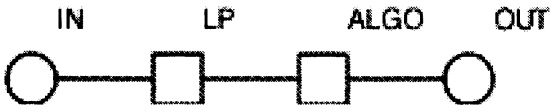

Figure 2. Simple mixed signal system

The third dimension of the implementation space is introduced by a vector of so called implementation possibilities, included in each functional node. The length of each vector corresponds to the number of functional equal implementations. This number varies with the different functions and may be limited by the designer according to corporation policy, customer demand etc.

Each implementation possibility contains a set of variables for nonfunctional properties that belong to the multi objective decision set. In the current project stage only the nonfunctional properties of ideal circuits and implementations are considered (for instance the active circuit area without wiring, the theroretical power consumption or the function related delay). But the model contains already variables for the difference between an ideal and a real implementation as far as nonfunctional properties are concerned. This concerns parasitic effects or equalizing extensions as well as routed connections between functional subsystems. If these nonfunctional properties of non-ideal effects cannot be statically determined by estimation methods, their determination is subject to simulation, also defined as dynamic estimation.

Environmental nodes contain only one implementation possibility which serves the sole purpose to specify the signals that are found at the system input(s) and the required signals at the system output(s). Another purpose of the environmental nodes is to propagate any knowledge from the specification into the implementation space model that cannot be expressed with signal classes or functions (nonfunctional

constraints for instance. An environmental node has no costs of nonfunctional properties.

A system configuration is again a two-dimensional representation of the system, after selecting one implementation possibility in each node. Now, in a configuration, we face the problem of signal conversion, when subsequent node implementations process signals of different classes. If converter nodes were to be inserted at this point, two different configurations would hardly have the same number of nodes.

Therefore each edge is always annotated with a converter shell regardless whether or not a converter is needed (Fig.3). This ensures a constant graph size and allows us to take the nonfunctional properties of wiring into consideration.

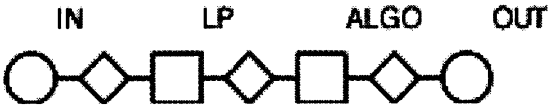

Figure 3. Converter shells in each edge

The embedded systemgraph is made of four different objects as shown in Table 3.

Table 3. Graph objects

Object	*Symbol*	*Use*
Source		Environment specification for input signals
Node		Functionally independent subsystem, specified according to Table 2
Edge		Signal transport. Is annotated with an container for different converter implementation possibilities
Sink		Environment specification for output signals

3.2 Analysis

The evaluation of a configuration passes three stages:

1. An implementation possibility is chosen in each block and its costs are determined.

2. The requirement of signal conversion is determined and zero costs (or equivalent to zero) are assigned to unneeded converters.
3. For each type of conversion (A/D, B/D, D/B) a implementation possibility is chosen and the converter-node assigned with its costs. The number of converter implementation possibilities is usually limited by timing constraints, otherwise the choice of the converter architecture again is subject to an optimization process.

Because of the complexity of the embedded system graph we chose to use heuristic metagraph algorithms for the implementation space analysis. Our first objective is an implementation of Simulated Annealing [1]. Afterwards Tabu Search [3] could be adapted alternatively.

Each system configuration corresponds to a node of a hypergraph that models its neighbor relations. A neighbored configuration is a configuration that differs in a single implementation possibility in one given functional node. Simulated Annealing starts at a given evaluated configuration and jumps with a certain probability to a neighbored configuration analyzing its costs. The neighbored configuration is chosen randomly by modifying the active implementation possibility in a randomly chosen functional node.

Depending on the start configuration and on the temperature of the algorithm, the behavior of the algorithm models different partitioning strategies.

For example *-find the partition with maximal digital domain-* or *-try to enlarge the analog domain-* are common strategies in mixed signal design.

3.3 Decision Theory

From the decision theory point of view the analysis and optimization of an implementation space is a single actor/multi objective decision problem with certainty and without risk [7]. Any nonfunctional property that can be evaluated by linear cost functions may be included in the multi objective set.

Nonfunctional properties and their associated constraints represent either contradictive or linked goals. Optimization concerning chip area and concerning delay is an example for contradictive goals. Improvement in one cost function leads to deterioration for the other. Chip area and power dissipation for example are linked goals. The optimal solution for such multi objective decision is usually defined as pareto optimum. Since Simulated Annealing randomly jumps from configuration to configuration, one cannot simultanously observe the development of each objective. Goal programming is used instead. First each optimization goal is optimized individually. Results are combined to the (only theoretically existing) goal vector.

The optimization algorithm starts again but now the complete objective set is compared to the goal vector using an lp metric. Metric elements may have additional weights to encourage optimization towards desired objectives. The pareto optimal solution can be approximated in this way, a solution should at least be element of the pareto set [8].

3.4. Example

Let us assume for the simple mixed signal system depicted in Fig.6 that an analog input signal shall be filtered with a low pass and then be processed by an algorithm. The algorithm's results shall leave the system as 16-Bit digital signal.

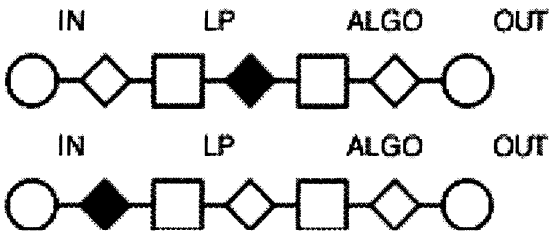

Figure 4. Configurations of the simple system

It is easy to see that the converter node after the algorithm will always remain empty. But, as shown in Fig.4, there exist at least two different configurations for the signal preprocessing.

1. The low pass is implemented with analog circuits, the first converter remains unneeded and the second converter is a B/D converter
2. The first converter is an A/D converter, the low pass is implemented digital and the second converter is unneeded.

Note that in reality there exist many more different configurations for this simple system. An analog low pass behavior can be implemented with at least ten different circuits, each with its own functional and nonfunctional properties. A digital filter can be implemented as one of four FIR circuits, one of eight IIR circuits or as FIR/IIR software realization on a DSP or on a microprocessor [2], again each with its own functional and nonfunctional properties.

Each configuration is multiplied by three according to the three conversion speeds: serial, weighing and parallel conversion. A digital FSMD implementation may have different space or timing requirements (nonfunctional properties) depending on scheduling and synthesis specifications.

Even this simple embedded system has a considerable implementation space that can hardly be explored without computer aid. But even for such a simple embedded system the implementation space exploration may be worthwhile to find the configuration that satisfies functional requirements, does not violate nonfunctional constraints and makes efficient useage of given resources.

4. Summary and Conclusion

A top-down design method for mixed signal systems was introduced, with main focus on signal processing embedded systems. Key feature of this Analog/Digital

Co-Design is the common consideration of analog and digital implementations in a common design space.

The design space is modeled with a formal graph representation. By annotation of costs for nonfunctional properties to each design alternative in choice the design space is transformed to a so called implementation space. The implementation space is modeled with the same graph representation, which is called systemgraph. The systemgraph is provided with cost values by means of estimation, simulation or synthesis. Different system configurations and their costs can be extracted from the graph. Together they represent a configuration hypergraph which can be optimized with probabilistic heuristics such as simulated annealing.

An example has demonstrated that a large implementation space can be found for even the most simple embedded system. It could not be handled effectively with non-automated or with iteratively operating methods.

We believe Analog/Digital Co-Design to be a promising approach to improve the quality and the level of automation in mixed-signal design.

Acknowledgements

This research project is partly funded by the Deutsche Forschungs Gemeinschaft under reference WA 357/14, where research is focused on the estimation of nonfunctional properties for signal processing circuits.

References

[1] E. Aarts and J. Korst. Simulated Annealing and Boltzmann Machines. Wiley & Sons Inc., 1988

[2] S. A. Azizi. Entwurf und Realisierung digitaler Filter. Oldenbourg Verlag, München, Wien, fünfte Auflage, 1990.

[3] R. Battiti and G.Tecchiolli. The Reactive Tabu Search. In ORSA Journal on Computing, 1993.

[4] Klaus Buchenrieder. Hardware/Software Co-Design, An Annotated Bibliography. IT-Press, Chicago, 1995.

[5] Christoph Grimm and Klaus Waldschmidt. „Repartitioning and technology mapping of electronic hybrid systems". Design Automation and Test in Europe 98 (DATE), Paris, France, February 1998

[6] Sanjaya Kumar, James Aylor, Barry Johnson and W. Wulf. The Codesign of Embedded Systems. A Unified Hardware/Software Representation. Kluwer academic Publishers, Boston, Dordrecht, London 1996

[7] Anatol Rapoport. Decision theory and decision behaviour: normative and descriptive approaches. Kluwer Academic Publishers, Dordrecht 1989.

[8] R.L. Yu. Mutiple-Criteria Decision Making. Plenum Press, New York,1985.

[9] Frank Heuschen, Christoph Grimm, Klaus Waldschmidt. „Modellierung des Implementierungsraumes im Analog/Digital Co-Design". In Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, 3. GI/ITG/GMM-Workshop, Frankfurt, Deutschland, Februar/März 2000. Tagungsband: ISBN 3-8007-2524-X, VDE Verlag, Berlin, Offenbach 2000

[10] Frank Heuschen, Holger Schmitt, Klaus Waldschmidt. „Formale Modellierung eines gemischt analog/digitalen Systems". In Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, 2. GI/ITG/GMM-Workshop, Braunschweig, Deutschland, Februar 1999. Tagungsband: ISBN 3-8265-4684-9, Shaker Verlag, Aachen 1999