# Synthesis on Programmable Analog Devices from VHDL-AMS

G. Doménech Asensi, J.S. Gómez-Díaz, J. Martínez-Alajarín, R. Ruiz Merino
Dpto. de Electrónica y Tecnología de Computadoras
Universidad Politécnica de Cartagena
Cartagena 30202, Spain
gines.domenech@upct.es

***Abstract*—Nowadays, the microelectronics market is characterized by an increasing complexity and integration, in particular in the field of application specific integrated circuits (ASICs) not only for digital but also for mixed-signal designs. The lack of a well defined methodology for analog synthesis, similar to the digital field means a serious drawback for analog and mixed signal design development. In this sense, the arise of VHDL-AMS is a recent evolution which promises to link analog design automation tasks into a coherent framework, in a similar fashion that digital design. In this paper, a tool to perform automated synthesis of analog systems, described in VHDL-AMS, into analog programmable devices is presented. The tool is focused to synthesise filters, wave-shaping circuits, amplifiers and in general most circuits supported by programmable technology. It is demonstrated with a practical example of a analog system composed by two filters and two controllable gain stages.**

## I. Introduction

Nowadays, the microelectronics market is characterized by an increasing complexity and integration, in particular in the field of application specific integrated circuits (ASICs). According to [1], the mixed-signal IC market has grown 15% to 20% per year since early 1990s. However, the development of appropriate synthesis methodologies to support mixed-signal ASIC designs is lagging and the analog parts of a typical ASIC still need to be manually designed. It has been reported [2] that with the advent of digital synthesis tools and semi-custom layout design techniques, analog ASIC blocks may now consume 90 % of the overall design time, while using only 10% of the silicon area. The lack of a well defined methodology for analog synthesis, similar to the digital field means a serious drawback for analogue and mixed signal design development. In this sense, the arise of VHDL-AMS, defined by the IEEE DASC 1076.1 standard [3], is a recent evolution which promises to link analog design automation tasks into a coherent framework , in a similar fashion that digital design. This framework should cover analog design hierarchy from design conceptualization to manufacturing and verification, also including digital circuits and interfaces to link both domains. From this standard, new CAD tools able to synthesise high level analog and mixed signal descriptions into electronic circuits are expected to appear in the next years. In this sense, in order to bring synthesis of these systems into such design automation framework, it is first needed to describe these circuits using a proper hardware description language. In the case of mixed signal circuits, VHDL-AMS is a suitable language to be taken as starting point in the synthesis process, just as VHDL currently represents the higher level steps in digital circuits design hierarchy. In this paper a methodology to synthesise analog designs into programmable analog devices is presented. It takes advantages of the VHDL-AMS potential to describe analog and mixed signal systems as starting point for the synthesis process. Destination technology is based on programmable devices. This allows the design cycle for analog systems to be similar to that used in pure digital designs. The advantages of automated digital design, which is used by thousands of designers with extraordinary results, are obvious. The structure of this paper is as follows: in section one, a presentation of the work is made; in section 2, the structure of the tool is presented; in section 3, a case study is shown; in section 4 the obtained results are commented; and in section 5, our conclusions are presented.

## II. Tool Overview

The use of tools which take VHDL-AMS as input code to generate some type of analog electronic implementations is evolving in recent years. A first approach was developed in [4], where VHDL-AMS inputs were translated into SPICE outputs. Other tools like VASE [5] translated VHDL-AMS behavioural-specifications of analog systems into op amp level net-lists of library components Tools based on architectural synthesis able to obtain SPICE output code were presented in [6] to build analog systems with feedback and in [7] to synthesise analog filters. Finally, other type of tools have been designed specifically just to refine and simplify, at high level, analog behavioural models written in

This work has been supported by Ministerio de Ciencia y Tecnología of Spain, under grant TIC2003-09400-C04-02.

1-4244-0088-0/06/$20.00 ©2006 IEEE

VHDL-AMS [8] In general, these tools were focused in high level synthesis, due to the intrinsic limitations of the low level one, given the performance degradation that occurs when automating analog layout process. The lack of a technology which could avoid such problems is only partly solved with the field programmable analog arrays (FPAAs), because of their limitation to synthesise complex analog designs. However, at op-amp level synthesis, programmable devices exhibit great advantages with respect to other technologies. In this context, electronic design automation tools can offer improved front-to-end solutions for analog design. Fig. 1 shows the general schema of the synthesis tool proposed in this paper. The conversion is performed in three main steps. In the first one, VHDL-AMS is translated into an intermediate code in raw format. In a second step, this code is translated into a set of script commands. Finally the script commands are compiled by a proprietary software provided by the FPAA vendor which produces the device configuration file. Therefore, in this paper we explain the structure of the modules which perform steps one and two, in which architectural synthesis from VHDL-AMS into a structural description is performed. The third step is a simple translation of formats, to produce a configuration file for the programmable device.

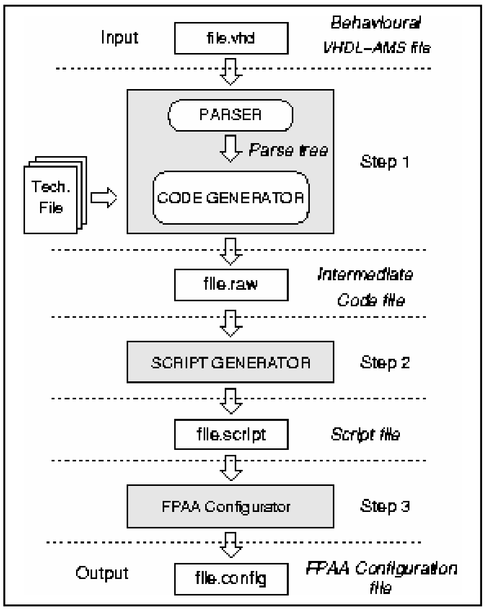

Figure 1. Synthesis Tool Structure

The structure of the first module, responsible for step one, is derived from the works done in [6], where the code generator has been modified to produce a format different from SPICE. Firstly, semantic and syntactic checks are performed on the input VHDL-AMS code and a parse tree id created. The structure of the parse tree generator has been developed according to [9]. This tree is then processed in order to obtain an output code, (intermediate code file). As any other synthesis tool, the code generator needs specification of technology employed for synthesis, which is supplied through a technology file compiled with the input file. This technology file contains description about the configurable analog blocks of the FPAA, their parameters, interconnections, etc. Thus, in step one, behavioural description of input file is translated into a structural description. This step is further detailed in section III. The intermediate code file is a structural description of the system. Basically, it includes both a list of configurable analog blocks used, as well as their configuration parameters, and a list of interconnections. This information is similar to that used to specify FPGA configuration, for digital systems. The second step translates this intermediate code file into a script file. At this level, we have structural descriptions based on architectural netlists (fig. 1). The translation is easily done by means of a direct mapping of each configurable analog block parameters and their interconnections into a new format. This step, detailed in section III, is needed to provide a link between architectural synthesis module (step one) and FPAA configuration software (step three). Finally in step three the script file is read by the FPAA configuration software, which produces the configuration file and programs the device. This last step is done using the software provided by the FPAA vendor, as a FPAA programming model is not available to users.

## III. Case Study

A topic application for an analog design is a data acquisition system. In this paper we have focused on an smart stethoscope, an analog system specifically designed to register and evaluate phonocardiogram sounds, as this is one of the hot topics in current research for medicine and surgery. The general structure of such system comprises an analog interface, an analog to digital converter and a digital processing part.

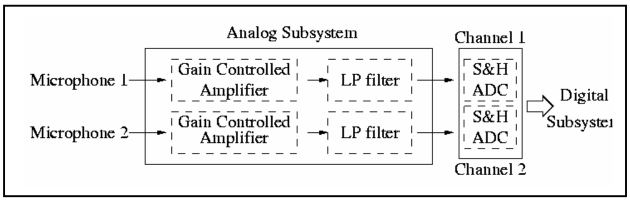

Figure 2. Mixed signal system layout

Fig. 2 shows a schema of the analog subsystem of the acquisition systems, where the two microphones are connected to both gain controlled amplifiers whose output are driven to respective low pass filters. Filter outputs are then converted into digital signal signals and processed in the digital subsystem. One of the input channels is connected to an stethoscope and the other to a pulsometer.

### A. High Level Analog Description

Let us focus in the analog part of the system. Starting from the higher analog levels, eqns. (1) and (2) represent the algorithmic level of the analog subsystem for a single input channel:

$$amp\_out = vin \cdot vcontrol \tag{1}$$

$$Vout(s) = \frac{K \cdot amp\_out(s)}{s^2 / \omega^2 + s / \omega Q + 1} \tag{2}$$

where *vin* and *vcontrol* are the input voltage from the microphone and the gain control voltage respectively, *vout* is the system output and *K*, *Q* and *ω* are filter parameters. These are the equations for a voltage controlled amplifier and for a second order low pass filter. In order to rewrite this description level in terms of VHDL-AMS, we need to create a pair *entity-architecture*. The *entity* describes the interface of the system, while the *architecture* describes the model itself. Fig. 3 shows the VHDL-AMS code written for the analog subsystem.

```
entity estethoscope is
 generic (K: real:= 1.0;     -- gain
  P: real:= 1.0;             -- gain
  Fp1: real:= 1.0e3;         --ch1 pole freq
  Fp2: real:= 2.0e3);        --ch2 pole freq
 port (terminal input1,input2, control1,
  control2, output1,output2: electrical);
end entity estethoscope;
architecture behaviour of estethoscope is
 quantity vin1 across input1 to electrical_ref;
 ...
 constant wp1: real:= math_2_pi*Fp;
 ...
begin
  amp_out1==vin1*vcontrol1;    --eq.1
  vout1==K*amp_out1'ltf(num1,den1);--eq.2
  amp_out2==vin2*vcontrol2;
  vout2==P*amp_out2'ltf(num2,den2);
end architecture behaviour;
end
```

Figure 3. VHDL-AMS description of the esthetoscope

In the *entity*, generic interfaces are used to define constant parameters. In this example they include constant gain values for K and P as well as corner frequency for both filters. Port interfaces are used to describe system inputs and outputs. The stethoscope analog subsystem includes two inputs and two outputs corresponding to both processing channels. The *architecture* is structured in two sections. In the declarative part, terminal, quantities and constants are defined. In the statement parts, the model of the system is described. This part includes two pairs of simple simultaneous statements. Each pair of sentences defines behaviour of a single stethoscope channel. The first sentence describes the controlled gain stage, as an analog multiplication. The second sentence describes a Laplace transform using the implicit quantity Q'LTF. This is a VHDL-AMS feature which allows a compact and clear notation for this type of systems.

### B. Synthesis Process

As explained before, the first step in the analog synthesis is the transformation of the VHDL-AMS behavioural description into a raw format structural description. This first step comprises two tasks: building a parse tree and generating an intermediate code. Fig. 4 shows part of the parse tree, corresponding to the two first simple simultaneous statements in the architecture statement part.

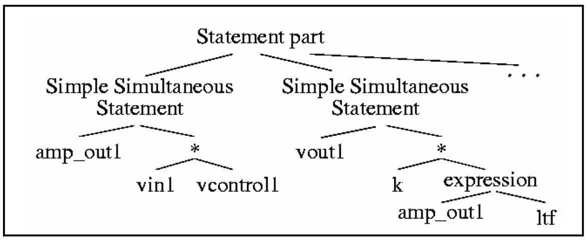

Figure 4. Structure of the parse tree

Each main branch of the tree describes the structure of a *simple simultaneous statement.* First statement is the analog multiplier, where quantity *amp_out1* in left term is equal to the multiplication of quantities *vin1* and *vcontrol1* in the right term of the sentence. In the other statement, quantity *vout1* in the left term is equal to the multiplication of constant *K* and an expression. This expression comprises the implicit quantity *amp_out1'LTF*, which is the Laplace transform of *amp_out1*, according to VHDL-AMS. Values for numerator and denominator of the Laplace transform, appear deeper in the tree structure, as two branches hanging form *ltf*, and are not shown for not to overdraw the figure.We can see how parse tree development is a suitable technique to transform behavioural descriptions into structural ones.

In this example we have chosen an FPAA from Anadigm [10] as programmable device to carry out the synthesis. This device has an I/O interface composed by four inputs and two outputs and a processing capability of four configurable analog blocks. Each one of these blocks can synthesise a set of analog, as well as an 8 bit analog to digital converter. These analog functions are represented as configurable analog modules (CAMs) inside a configurable analog block. Fig. 5 shows the intermediate code generated after step one by the synthesis tool.

```
VHDL-AMS-Compiler
Chips=1
Name=estethoscope.vhd
CAMS
InputCell=1;0;0;0;0;150;50
InputCell=2;0;0;0;0;150;50
InputCell=4;0;0;0;0;150;50
InputCell=3;0;0;0;2;150;50
OutputCell=1;0;100
OutputCell=2;0;100
Multiplier=1;0;2
FilterBiquad=1;0;0;1;1;1;0.707;1;1
Multiplier=2;0;2
FilterBiquad=2;0;0;1;2;1;0.707;1;1
Interconnections
InputCell1(1)=Multiplier1(1)
InputCell2(1)=Multiplier1(2)
Multiplier1(1)=FilterBiquad1(1)
FilterBiquad1(1)=OutputCell1(1)
InputCell3(1)=Multiplier2(1)
InputCell4(1)=Multiplier2(2)
Multiplier2(1)=FilterBiquad2(1)
FilterBiquad2(1)=OutputCell2(1)
end
```

Figure 5. Intermediate code

First lines set general settings of the file and the FPAA chip. A second group of lines sets parameters configuration of all configurable analog blocks used. The last section of the file describes the interconnection between CAMS and I/O blocks. It is important to note that no connection between configurable analog blocks needs to be specified. Each sentence of this list comprises an output in the left term and an input in the right one. Tab. 1 shows the list of parameters of some configurable blocks used in code shown in Fig. 6. System works setting only specifying only connections between all CAMs in the chip. How these CAMs are arranged in one or more configurable analog block is a process done in third step of synthesis process (Fig. 1).

TABLE I. CONFIGURABLE ANALOG BLOCK PARAMETERS

| Name | Options |
|---|---|
| **Multiplier** | **Cell Number**<br>**Sample & Hold**: 0 off, 1 input X<br>**Multiplication Factor** |
| **Biquadratic Filter** | **Cell Number**<br>**Filter Type**: 0 LP, 1 HP, 2 BP,3 band stop<br>**Input Sampling**: 0 phase 1, 1 phase 2<br>**Polarity:** 0 inverting, 1 non-inverting<br>**Corner Frequency (kHz)**<br>**Gain**<br>**Quality Factor**<br>**DC Gain**<br>**High Frequency Gain** |

## IV. RESULTS

Configuration of the FPAA device is shown in Fig. 6. In the figure we can see the arrangement of configurable analog blocks in the FPAA. It uses two multipliers, as voltage controlled gain stages and to biquadratic filters. Multiplier I is connected to inputs 1 and 2. Input 1 (pins 11 and 12 in the device) is connected to the microphone and input 2 (pins 09 and 10) is connected to the voltage controlled gain. Multiplier output is connected to filter I through net *n5*.

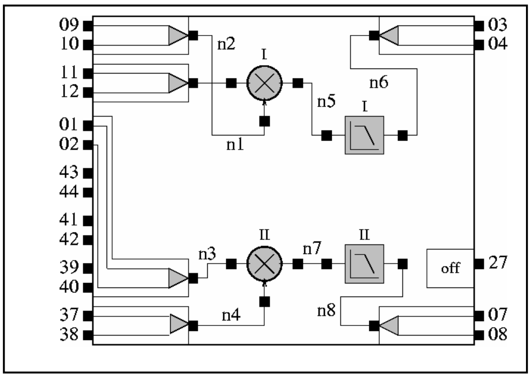

Figure 6. FPAA Configuration

Filter output is finally connected to output 1 (pins 03 and 04) through net *n6*. Arrangement of analog channel 2, containing multiplier II and filter II is similar to channel 1. Fig. 7 shows input and output signal of the analog system. Input corresponds to a systolic sound registered by microphone one of the stethoscope, while output is this same signal amplified, sampled and filtered. The controllable gain value has been set to 2 units.

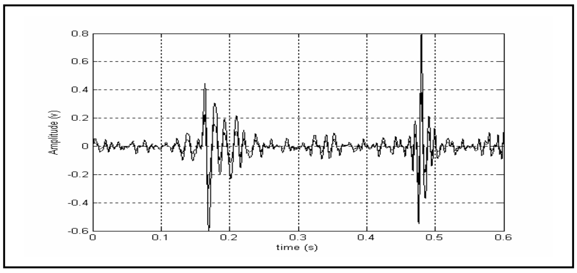

Figure 7. Simulation Results

## V. CONCLUSIONS

In this paper a methodology to synthesise analog designs into programmable analog devices has been shown. It takes advantages of the VHDL-AMS potential to describe analog and mixed signal systems as starting point for the synthesis process. Destination technology is based on programmable devices, which allows the design cycle for analog systems to be similar to that used in pure digital designs. The synthesis process is done in three steps, where the architectural synthesis is carried out firstly, in order to translate behavioural descriptions into structural specifications. A second steps links the structural description intermediate code with a proprietary software which performs FPAA programming. The tool has been demonstrated with the synthesis of the analog subsystem of an electronic stethoscope. Although currently available technology offers small sized FPAA devices, we hope in a near future bigger devices will be developed, able to synthesise analog designs of higher complexity.

## REFERENCES

[1] G.E. Gielen, R. A. Rutenbar "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", Proceedings. of the IEEE, 88, nº 12,(2000), 1823-1852

[2] E.S. Ochotta, R.A. Rutenbar, L.R: Carley "Synthesis of High Performance Analog Circuits in ASTRX/OBLX" IEEE Transactions on Computer Aided Design, 15 (1996) 273-294

[3] "Definition of analog and mixed signal extensions to IEEE Standard VHDL". IEEE Standard 1076.1-1999

[4] S. Li, B. Okoon, M. Hella, M. Ismail and M. Rubeiz "The Implementation of a VHDL-AMS to SPICE Converter", Analog Integrated Circuits and Signal Processing, vol 20, nº3,1999 203-211

[5] R. Vemuri, N. Dhanwada, Adrian Nunez, P. Campisi; ``VASE: VHDL-AMS Synthesis environment Tools for Mixed-Signal Systems"; Proc. EETimes Conference on Analog and Mixed-Signal Applications; pp. 1C77-1C84, July 1997.

[6] G. Doménech, T.J. Kazmierski, R. Ruiz, and J. D. Ruiz "Architectural synthesis of high level analogue VHDL-AMS descriptions using netlist extraction from parse trees", Electronic Letters, vol 36 nº20 (2000) 1680-1682

[7] T. J. Kazmierski, F. A. Hamid, "Behavioral modelling of RF filters in VHDL-AMS for automated architectural and parametric optimization " *Proc. of the 2004 Int.l Symp on Circ. and Sys*, May 2004.

[8] K. Zeng, S. A. Huss, "RAMS: A VHDL-AMS Code Refactoring Tool Supporting High Level Analog Synthesis" IEEE Computer. Society. Annual Symp on VLSI.New Frontiers in VLSI Design, 2005

[9] A.V. Aho, R. Sethi and J. D. Ullman "Compilers: Principles, Techniques and Tools" Addison-Wesley, Massachusetts, USA. 1986

[10] "AN221E04 User Manual" Anadigm, Inc. California, USA.