Simulating Field Programmable Analog Devices

Sandro Wefel* Paul Molitor
Institute for Computer Science
Martin-Luther-University Halle
06099 Halle, Germany

Manfred Rost
Department of Physics
Martin-Luther-University Halle
06099 Halle, Germany

Abstract

FPGA architecture forms the basis for a new generation of programmable analog circuits allowing to easily realize analog functions by hardware. In order to make this technology applicable powerful tools for synthesis and simulation have to be made available.

This article presents an approach to simulate programmable analog devices. Structural information and quantitative features are extracted from a given configuration of the programmable analog device. After various simplifications of the netlist which are necessary to avoid large simulation times a netlist for SPICE is generated.

Experimental results show that the simplified netlists generated by our tool are adequate to obtain good simulations which approximate the actual behavior of the programmed analog devices. The benchmark set comprises filters, signal generators and signal amplification functions.

1. Introduction

Analog devices close the gap between natural environment and digital systems. Most sensors convert non-electrically analog measurement values from the natural environment (pressure, temperature, concentration) into electrically analog signals. The analog signals have to be converted into digital signals by analog-digital-converter after they have been preprocessed on the analog layer, as e.g. amplified, filtered, current-voltage-converted, and undesirable disturbance have been removed. All these operations are executed by analog signal processing.

In the design of small digital systems programmable devices, as e.g. Programmable Logic Arrays (PLAs) or Field Programmable Gate Arrays (FPGAs) have been used for reasons of costs and economic efficiency from the very beginning. For the same reasons there is a need for programmable analog devices.

In 1994, a first purchasable system of several hardwired programmable basic cells called TRAC (Totally Reconfigurable Analog Circuit) has been presented. The set of the basic cells available in the TRAC system consists of a complete base, which theoretically allows the design of every analog signal processing unit [3]. However, the hardwired connection between the cells restricts the functionality and the application of the TRAC system very much.

Another system, which has been recently developed, is the FPAD (Field Programmable Analog Device) ASB from the Fraunhofer Institut Mikroelektronische Schaltungen und Systeme Dresden, Germany. ASB 100 is based on FPGA technology known from the digital area [4]. The FPGA technology used makes the ASB 100 device very powerful. Unfortunately, (automatic) synthesis and simulation tools are not available, till today.

In this paper, we present an approach for simulating programmable analog devices based on FPGA technology such that the simulation time is acceptable and the simulation results are good approximations of the actual behavior of the programmed analog device.

The paper is structured as follows. In Section 2 we present the ASB 100 device in more detail. Section 3 concentrates on the extraction of a simplified netlist of a programmed ASB 100 device. This netlist is passed to the SPICE system. Experimental results discussing the quality and efficiency of the simulation are presented in Section 4.

2. Analog Devices

The FPAD ASB 100 device from the Fraunhofer Institut Mikroelektronische Schaltungen und

*email: wefel@informatik.uni-halle.de

1TRAC is a registered trademark of ZETEX Semiconductors
**Systeme Dresden** is based on FPGA-technology. A digital Field Programmable Gate Array consists of a central core with digital functional blocks (programmable multiplexer or combinatorial logic), a programmable connection matrix between these blocks, and connectors to the input/output-pads of the device. Substituting the digital function blocks by analog blocks results in a Field Programmable Analog Device (FPAD). The structure of the FPAD is shown in Figure 1. It consists of three identical blocks, the building blocks, containing a wide range of basic analog circuits. The connection between these blocks is programmable because of switches. The building blocks together with a Voltage Controlled Oscillator (VCO cell) form the analog core of the circuit. The peripherals of the ASB 100 device consist of reference power supplies, input/output-pads, programming interface, and some more additional cells.

A building block of the ASB 100 device is presented in Figure 2. It consists of two MAI-cells which are used for analog multiplying, adding and integration, an additional adder and multiplier, passive elements as e.g. resistors and capacitors which are placed in the RESCAP subblock, a comparator which is used for signal depending path switching, and a six-bit analog/digital converter. For a detailed description, please refer to [2].

The connectors in the building blocks and between the blocks and the periphery are configurable. As different resistors and capacitors are available out of a discrete range, resistors and capacitors can be realized by connecting the basic elements in series or in parallel. Thus, the ASB 100 device makes available programmable resistors and capacitors. There are 5 modes for the power state (one power off and four power on modes) of the active elements allowing different maximal output currents.

Flip-Flop chains store the configuration of all the elements and connectors. In all there are ten chains which are programmed using a multiplexed input.

**Figure 1. Structure of FPAD ASB 100**

**Figure 2. Components of Analog Core.**
3. Simulation

The simulation procedure of the FPAD ASB 100 is shown in Figure 3. The input of the simulation is given by the ten bit vectors which determine the configuration of the programmed device. These data must be converted into a SPICE netlist. This conversion can be done in a straightforward manner. However, the netlist generated is very large as it contains all the elements and connectors available on the ASB 100 device. The netlist must be made smaller by simplifications and abstractions in order to speed up the SPICE simulation. The problem is to do this without making worse the simulation quality.

We will show by experiments (see Section 4) that replacing the switches by wires with respect to the programmed configuration of the switches and removing parts not (directly) needed are simplifications which do not dramatically make worse the simulation quality while allowing very fast simulation runs.

3.1. The unreduced netlist

The first step is to build a data structure containing a description of the whole circuit FPAD ASB 100. All elements of the device have an entry in that structure. Of course, the programmable characteristics of the elements are not included yet. The data structure consists of a list of subnets and a list of elements. A subnet is a part of a net which contains all elements which are connected. The element list represents the electronic elements, i.e., adder, multiplier, resistors, capacitors, switches, and so on. Each subnet and each element obtains a symbolic name.

In the next step the actual configuration which is specified by the ten bit vectors mentioned above is mapped on this structure. The bits of these vectors have informations on the switching states, the values of the resistors and capacitors, and the power mode of the active elements.

3.2. Netlist reduction

The next step is to make sparser the netlist by simplification. First, turned off elements are removed as it is possible to set an element in an actual configuration as turned off. A capacitor can have a value of 0 \( \mu F \), a resistor an infinite value, an active element can be powered off, and a switch can be at open state. All these elements are removed without any further consideration. A small error was made because turned off elements have a remaining resistance and capacity and could affect the rest of the circuit.

Another reduction is the combination of subnets. If two subnets are connected by a closed switch then these subnets looks like one subnet. The closed switches are removed and the corresponding subnets are combined. This simplification causes an error, as the remaining resist of closed semiconductor switches is ignored.

![Figure 3. Stages of Simulation](image)

![Figure 4. Remove unnecessary elements.](image)
the functionality of the circuit this is a superfluous element.

3.3. Generating the simulation input

After having applied the reduction steps the information generated is converted in a SPICE netlist using the symbolic names of the elements and the wires. Each element which has not been removed is put on. Such an entry looks like, e.g.,

\[ R_{R1} \text{ N0001 N0002 100k} \]

which stands for a resistor with symbolic name \( R_{R1} \) which is located in the nets with symbolic names \( \text{N0001} \) and \( \text{N0002} \) and which has a value of 100k Ω.

Only now, library dependent information is introduced in the netlist. Circuit intern macro elements as e.g. operational amplifiers must be replaced by library elements of the simulation program. For our experiments we have used common macro descriptions of all active elements without adjusting them to the basic cells used in the FPAD ASB 100. This will cause some divergence. (Of course, in industrial application precise macros for the basic cells of the FPAD should be used.)

4. Experimental results

Two different types of analog circuits have been used to validate the performance of the proposed simulation approach. The first serie consisted of filters (high-, low- and bandpass) from first order up to sixth order. The high order circuits have been constructed by cascading up to three biquadratic filters [3]. The second test serie consisted of signal generators, i.e., sawtooth, rectangle, and sinus waveform generators.

4.1. Filter circuits

Figure 5 shows the simulation results of an AC-analysis \(^2\) of a netlist generated from the ASB 100 device programmed as lowpass filter. The simulation is compared to the measured data of the ASB 100 device. Since AC-analysis is a linear analysis and the input voltage was set to 1 V ignoring any overriding of the maximum output, the output is the same as the gain (or attenuation) of the circuit. But nonlinear behavior or distortion in range near the supply voltage are ignored. Therefore AC-analysis could generate an output voltage which differs from the measured one, i.e. the output could be above the supply voltage. However note that signal transmission ratios from lower to higher frequencies can be adequately analyzed. Figure 5 also illustrates that the simulation ratios for a first order lowpass filter efficiently approximate the actual behavior of the device for a frequency of 60 kHz and more.

In Figure 6 is shown a sixth order lowpass filter. A control voltage modifies the signal transmission. Eleven different voltages are tested. The general behavior for simulation and measure are the same, the higher the voltage the higher the passing frequency. Biquadratic filters use analog multipliers for signal feedback which comprise some elements modelled by ideally elements in our simulation. This causes the differences between simulation data and measured data for the sixth order lowpass filter for the reason that the sixth order lowpass filter has been built by cascading three biquadratic filters and errors in the modelling of one filter are raised by connecting the filters in series.

4.2. Waveform generators

As example for signal generators a sinus waveform generator for the ASB100 is shown in Figure 7. The

\[
\begin{align*}
\text{Measure} & \quad \text{Simulation} \\
U_{\text{in}} \text{Voltage [V]} & \quad 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4 \\
VCO Frequency [kHz] & \quad 0, 50, 100, 150, 200, 250, 300
\end{align*}
\]

Figure 7. Voltage Controlled Oscillator.

Figure 8. VCO Output Frequencies of ASB100.

\(^2\)Alternate-Current-Analysis tracks the signal ratio between output and input signal over a given frequency range.
output frequencies for a control voltage between 0.1 V and 1.4 V are displayed in Figure 8. The frequency raising gradient of simulation matches the measure up to 1 V. For higher frequencies the real circuit has a non linear raising in contrast to the simulation which is shown in Figure 8. This seems to be caused by the same circumstances as in filter-circuits.

We have used PSpice of MicroSim Corporation for simulation. The simulation times on a Pentium PC are only few seconds. The computation of the reduced netlists requires less than 2 seconds.

5. Conclusion

An approach for simulating an analog programmable device has been presented. The algorithm works for a concrete device configuration using a bit vector as input. The method uses intensive reduction steps to extract simplified structural information and quantitative features from the given configuration. The simulation time for netlist creation is less than 2 seconds for the FPAD ASB 100 device handled here.

The efficiency of the described approach has been evaluated on different circuit examples. The system performs well on generator circuits and filters of lower order.

We have to investigate in future work whether the errors made in the simulation of filters of high order and complex circuits are mainly due to the library used during simulation which has not been adjusted to the basic cells of the FPAD ASB 100 device or to the simplifications made during the generation of the SPICE netlist.

References


