EDP Player

A Design Assistant for Procedural Design Automation of Analog Integrated Circuits

Matthias Schweikardt
Reutlingen University
72762 Reutlingen, Germany
matthias.schweikardt@reutlingen-university.de

Husni Habal
Infineon Technologies AG
85579 Neubiberg, Germany
husni.habal@infineon.com

Jürgen Scheible
Reutlingen University
72762 Reutlingen, Germany
juergen.scheible@reutlingen-university.de

***Abstract*—In this demonstration, we present the EDP Player, a tool for procedural circuit design automation of analog integrated circuits.**

I. Introduction

In this demonstration, we address circuit design automation of analog integrated circuits. Circuit design designates the conversion of a functional specification into a sized schematic. Procedural automation means, that the knowledge-based strategy of human experts is captured in an executable script, which makes it reusable. We call this principle EDP (**E**xpert **D**esign **P**lan). An EDP can cover different performance parameters, technologies and topologies. This paradigm is well-known from the domain of layout design of analog integrated circuits, where it is widely used by so called layout generators (e.g. PCells). With the EDP principle we want to provide an equivalent automation approach in the circuit design domain.
In order to realize the EDP principle, a tool for creation and execution of plain EDPs was created in the Cadence® Virtuoso® Design Environment (see II). The tool provides a preliminary version of an instruction set tailored to the typical manual analog circuit design flow, called EDPL (EDP-Language), based on Cadence® SKILL®. The tool has been utilized for several examples, e.g. the automated design of a miller operational amplifier and the automated creation of variants of a smart power IC (see III). The tool itself was developed using SKILL as well.

II. EDP Player Flow

The EDP Player can be started for every ADE XL view, which is a standard simulation environment in Cadence Virtuoso. The tool supports three different modes. The mode *Execute only (EDP given)* can be used if a given EDP should be applied on a circuit. In a wizard, the desired EDP can be selected first, which are loaded in background from a database. In a second step, the parameters and objectives of the ADE XL and EDP can be mapped. If they match by name, no mapping is needed. As a last step, the parameter and objectives ranges need to be specified. Afterwards, the EDP is executed and comes up with a result, which can be back annotated to the ADE XL view. The second mode is *Development (from scratch)*. In this mode, the designer can create his/her own EDP in the Command Interpreter Window or any text editor. The commands needed for EDP creation (retrieving and altering of parameters, executing simulations,...) can be easily looked up in a HTML documentation. Subsequently, the designer can save the EDP to the EDP database. The last mode, *Development (EDP given)*, can be used if the designer wants to adjust or expand a preexisting EDP.

III. Application Examples

The tool was used to automate sizing of a miller operational amplifier, which consist of eight devices. In a first step, expert knowledge was extracted from the circuit by performing analytical calculations. In a second step, this expert knowledge was realized in an executable EDP. The EDP consists of 99 commands. Within the EDP a total of eight objectives are considered, e.g. slew rate SR and phase margin PM. Furthermore, the EDP also checks that all transistors are in saturation region. The EDP returns a correct parameter set for a broad range of objectives. The execution time is, dependent on the objectives, between $15\,\mathrm{min}$ and $29\,\mathrm{min}$. Moreover, the tool was used to size a smart power IC, which consists of a DMOS and its control circuit. The objectives that need to be varied are e.g. the on-resistance $R_{\mathrm{DS(ON)}}$ or slew rate SR. Here, a given, final IC should be altered, that different product variants can be created automatically. Therefore, only some parameters, respectively eight, in the circuit need to be adapted. Altogether seven different EDPs were implemented for different use cases. Dependent on the use case, the EDP consists of 2 up to 27 commands and the execution time is between $6\,\mathrm{min}$ and $53\,\mathrm{min}$.

IV. Summary and Outlook

The methodology realized by the EDP Player proves to address real life design problems. Within above shown application examples, the EDPs always return a correct result. The coding of the above shown examples is comparatively simple (< 100 commands). The execution time is within reasonable range. The tool presented here contains only some basic analog design commands, hence the EDP language scope needs to be enhanced in the future.

V. Acknowledgements

We want to thank Infineon Technologies AG for providing the smart power IC circuit example.