IP-based methodology for analog design flow: Application on neuromorphic engineering

T. Levi , N. Lewis, J. Tomas, P. Fouillat
University Bordeaux 1, IMS Laboratory (CNRS/ENSEIRB), Talence, France
noelle.lewis@ims-bordeaux.fr

***Abstract*—Analog and Mixed Signal design flow has to be improved. In a specific application, neuromorphic engineering, we propose a definition of the analog IP (Intellectual Property) content and the structure of an IP-based library. The case study consists in the neuron-level integration of a complete system that emulates spiking neural networks. A reuse methodology based on the IP concept is developed and we show how it can be used to accelerate the design cycle of the next ASIC generation.**

I. INTRODUCTION

The main objective of this work is to improve the design flow of analog circuits. Purely digital systems profit from standardized tools and methods which allow automatic synthesis. While the trend is, for many years ago, to integrate both digital and analog circuits on the same chip, to build a mixed System on a Chip (SoC), the analog part is always the most costly in terms of design effort. This is due to the lack of formalism and automation in the analog design flow. Circuit sizing, layout and routing are the basic tasks but complex in the analog domain, because of the number of design parameters and the close influence of technology variations on these parameters. As a consequence, when the system-level specifications or the technology change, the designer has often to re-design the circuits from the bottom; system-level tasks like architecture exploration are also difficult to realise.

The main idea here is to re-use the accumulated design knowledge which could be illustrated by the IP (Intellectual Property) concept. This concept exists and is well defined in the digital domain: it may be purely virtual like in soft IP or include a hardware synthesised part like in hard IP [1]. In this paper, we propose a definition of an analog IP, which could be considered as hard IP, and show how to develop an IP-based library used to accelerate the design process of analog systems. This is applied in a specific domain which is neuromorphic engineering, where the developed ASICs must mimic the complex electrical behaviour of biological neurons.

After giving details on this particular application domain, this paper presents an IP-based methodology for analog design flow. Two application examples are shown as an illustration of this flow.

II. APPLICATION DOMAIN

Engineering of neuromorphic integrated systems is a research field where microelectronics encounters biology. The link between both is realized by computational neurosciences which model and emulate a part of brain activity. Different levels of modelling exist from the neuron physiology to the plasticity of large neurons networks. One issue is to have the adequate simulation system that implements those models; that is the role of neuromorphic engineering [2], [3]. Important features of such systems are re-configurability, observability and also real-time running, especially for hybrid (real/artificial) experiments.

From the microelectronic point of view, one solution is to design analog ASICs for the real-time computation of neurons activity and to digitally control the connectivity between these neurons.

To design such neuromimetic ICs, we chose the Hodgkin-Huxley neuron model, which is one of the more biologically realist [4]. Many neuromimetic ASICs have been designed by our team [5] [6] since 1993. Figure 1 shows the system hierarchy of the ASICs. The *cell level* corresponds to the implementation of each mathematical function building the neuron model; these cells are composed of typical analog blocks like differential amplifiers, OTAs, current mirrors, current-mode multipliers, etc. This level has been chosen as the basic IP-level.

Now, the trend is to integrate more and more complex neural networks, on the basis on recurrent primitive analog blocks. At the beginning of the design process, the specifications are given by neuroscientists and biologists; they consist mainly in parameters controlling the *block level* but also in some data related to the network topology.

978-1-4244-2332-3/08/$25.00 © 2008 IEEE

Due to the increasing complexity of networks to be implemented, it becomes crucial to maximize the re-use of previously designed blocks.

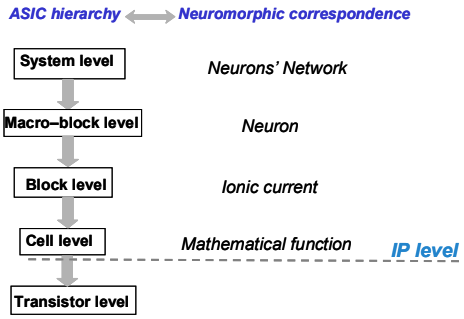

Figure 1: Hierarchy correspondence

III. IP-BASED LIBRARY

A. *Definition of the analog IP [7]*

Defining the IP content is the first important task because the IP concept is the base of design re-use. Some works already exist towards the re-use of AMS blocks [8] [9] [10].

First, our IP should give a precise characterization of already designed block; second, it has to be described with adequate representations or models, consistent with the design levels, to enable the easy re-use of the block, along the complete design flow.

Table I shows the different descriptions (or *views*) that are embedded in our IP blocks. All these views have the same terminals and the same symbol.

TABLE I. IP-AMS HIERARCHICAL DESCRIPTION LEVELS

View name	Description / Role
Symbol	visualizes the function
Connectical	verifies connection between blocks (Verilog-A)
Functional	models ideal electrical behaviour (Verilog-A)
Behavioural	models non-ideal electrical behaviour, extracted from schematic (Verilog-A)
Schematic	transistor-level schematic
Layout	
Datasheet	See below …

For the *connectical*, *functional* and *behavioural* views, we use Verilog-A language. These views are useful for multilevel simulations especially in the verification phase of the design process. The *functional* view describes the ideal equations of the function to be implemented. The *behavioural* view is more detailed indeed there are refined equations which fit the schematic behaviour.

One important point is to have a fluent and coherent design flow, and logical and mathematical links have been established between the different views.

The *datasheet* view is the most important view for the research of the corresponding IP. This view is the main point of the re-use methodology. It contains information about the design (the technology, the supply voltage, the terminals, the area of the layout), the circuit specifications (in our case, dynamic range of inputs) and information for the circuit re-usability (characterisation procedure, links between the functional model and the behavioural model equations, Monte-Carlo robustness evaluation).

B. *Data-base implementation and exploration*

All characterized IPs have been collected in a MySQL data-base. This data-base implements the IPs content and the hierarchical links between IPs. Each mathematical neuronal function is associated to one or several IP blocks.

In the ASIC design flow, the data-base is automatically explored according to the system specifications which are given by the neurobiologists. Then these specifications are converted in electrical ones. The exploration method is a Top-Down one, from the *macro-block level* to the *cell level*; and an IP is selected if its dynamic input ranges can cover the targeted specifications. If we have different corresponding IPs, we take the closest to the specifications according to a simple cost function. The exploration requests are automatically created in PHP.

Figure 2 represents this system design flow.

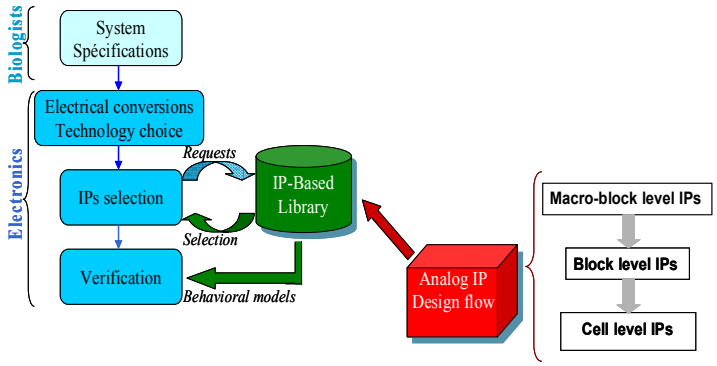

Figure 2: IP-based design flow

C. *Results of the exploration*

At the final step, a diagnosis is returned which quantifies the possibility to re-use IPs for a new ASIC project. In a case of total re-use, the IPs content is recovered by the designer with the system netlist. Then he can use this netlist to verify the system with the behavioural views. If an IP could not be found, the software clearly identifies the error and the designer can use the datasheet view to find a way to adjust some design parameters and re-design a correct IP.

IV. EXAMPLE 1: ASIC AUTOMATIC DESIGN

A. Example description

All of these applications, corresponding to realistic design projects, are based on the IP-based library. This first application describes the automated design of a neuromorphic ASIC.

As illustrated on Figure 1, the ASIC is composed of neurons which are defined by different ionic currents (Sodium Na, Potassium K, Calcium Ca and Leak current) which are built by some elementary functions (activation, inactivation, kinetic, power and output level).

B. Specifications

The ASIC includes two types of neuron: Neuron 1 is an excitatory neuron and Neuron 2 is an inhibitory one. Neuron 1 has an adaptation current (Ca type) in addition. The model card of each neuron before electrical conversion is summarized in Table II.

TABLE II. PARAMETERS OF THE SYSTEM SPECIFICATIONS

Currents	Voffset (mV)	Vslope (mV)	Tau (ms)	Vequi (mV)	gmax (S)
Neuron 1					
Na	-37	7.2	0.03	50	110µ
K	-37	11.38	3	-100	22µ
Ca	-35	11.4	8	-700	50µ
Leak					220n
Neuron 2					
Na	-41	5.7	0.08	80	70µ
K	-31	13.81	5	-120	13µ
Leak					310n

C. Results

The data-base automatic exploration process has been tested for this specifications set. Table IV defines the results of the exploration according to the legend of Table III: a grey and black square respectively identify an existing IP and a non existing IP in the database.

TABLE III. PARAMETERS OF THE SYSTEM SPECIFICATIONS

No need of IP	IP exists	No IP

It can be observed on Table IV that all IPs are found in the data-base; we have a total reuse and the netlist of the system is given with the different needed IPs. The verification process can then begin.

TABLE IV. DIAGNOSIS OF THE EXPLORATION 1

Neuron 1

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						

Neuron 2

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						
Ca						

D. Verification

The Bottom-Up phase checks if the proposed architecture really meets the specifications. Indeed the behavioural and functional models are used to perform a multi-level simulation of the entire ASIC.

All the blocks have been validated using the analog simulator *Spectre* under *Cadence* environment. Due to the huge number of components that compose the chip, it is not possible to perform transistor-level simulation of the final chip. Furthermore, the neural activity is a low frequency activity, and the neural activity has to be simulated for at least tens of ms as shown in Figure 3.

We decided as a consequence to mix transistor level, behavioural, functional and connectical descriptions of the different blocks to validate their connectivity and ensure the functionality of the whole design. As an example, a simulation for one neuron lasts 2'17" using *Verilog-A* description and 14'56" using transistor level description. Both simulations give the same neural activity as depicted in Figure 3.

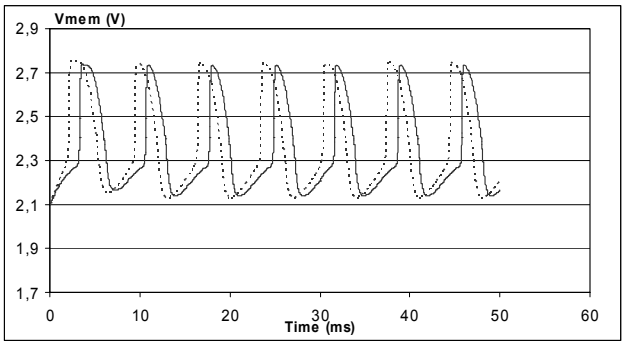

Figure 3: electrical activity of a spiking neuron (dashed line corresponds to Verilog-A description and continuous line to transistor level description).

To perform the validation of the whole circuit, we use the *Hierarchical Editor* of *Cadence* environment. For each simulation, only one block is described at the transistor level while the others are described using behavioural description.

V. EXAMPLE 2: EXPLORATION ISSUE AND HELP FOR RE-DESIGN

This second application describes the treatment of an exploration issue. It defines the given help for the designer, to create a new IP which corresponds to the specifications from the reuse of an existing IP.

In this example, we deliberately introduce an issue in the parameters specification for the neuron 1 in the sodium current.

A. Results

Table V represents the diagnosis of this new set of specifications and, as it was expected, an issue appears in the inactivation function of the sodium current. At this step, the designer has to solve this issue. To do it, one help is done.

TABLE V. DIAGNOSIS OF THE EXPLORATION

Neuron 1

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						

Neuron 2

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						
Ca						

B. Help for the reuse

There are three different requests and then three different exploration issues: technology, supply voltage and parameters specification.

When the issue is about the technology or the supply voltage, we developed a resizing methodology [11] to solve it.

When the issue is about the parameters specification (that's the case), a cost function finds the IP with the closest specifications. The designer uses its datasheet view and tunes its design parameters to create a new IP with required specifications. This new IP is then characterized; to be included in the database. Another exploration can finally be performed.

C. New exploration

Table VI represents the new diagnosis of this last exploration, and all the IPs are now found. The issue is solved and the designer can design and verify the new system.

TABLE VI. DIAGNOSIS OF THE EXPLORATION 2

Neuron 1

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						

Neuron 2

	Activation	Inactivation	Kinetic 1	Kinetic 2	Power	Output level
Na						
K						
Leak						
Ca						

VI. CONCLUSION

In this paper, a case-study in analog IP re-use is exposed. The questions that have been developed are: analog IP definition, IP-based library, IP-based system design and help for the reuse. To accelerate the time of design of our system and improve the design re-use, we define an IP-based library of analog elementary functions. This database allows us to build more rapidly the next generations of ASICs. The library associates behavioural and schematic views of all blocks at all hierarchical level, to be able to perform simulations of the whole chip in a reasonable CPU time.

REFERENCES

[1] *Model taxonomy version 2.1 (SLD 2 2.1)*, VSI Alliance™, 2001

[2] S. Saïghi, Y. Bornat, J. Tomas, S. Renaud, "Neuromimetic ICs and System for Parameters Extraction in Biological Neuron Models", Proceedings ISCAS 2006, pp.4207-4210, Island of Kos, Greece, May 2006.

[3] Y. Bornat, J. Tomas, S. Saïghi, S. Renaud, "BiCMOS Analog Integrated Circuits for Embedded Spiking Neural Networks", Proceedings DCIS 2005, Lisbon, Portugal, November 2005

[4] A.L. Hodgkin, A.F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", Journal of Physiology, 1952

[5] S. Renaud, J. Tomas, Y. Bornat, A. Daouzli, S. Saïghi, "Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks", International Symposium on Circuits And Systems 2007, pp 3355-3358, New-Orleans, USA, May 2007

[6] http://neuromorphic.ims-bordeaux.fr

[7] T. Levi, N. Lewis, J. Tomas, P. Fouillat, "IP-based design for analogue ASICs: A case study", IP-based SoC Design Conference 2006, Grenoble, France, December 6-7 2006, pp. 135-139, May 2007

[8] Z. Li, L. Luo, J. Yuan, "A Study on Analog IP Blocks for Mixed-Signal SoC", Proceedings ASIC, pp.564-567, Beijing, China, October 2003

[9] I. O'Connor, F. Tissafi-Drissi, G. Révy, F. Gaffiot, "UML/XML-based approach to hierarchical AMS synthesis", Proceedings FDL 2005, Lausanne, Switzerland, September 2005

[10] N. Martínez Madrid, E. Peralías, A. Acosta, A. Rueda, "Analog/Mixed-Signal IP modelling for design reuse", DATE Conference, Munich, Germany, 2001

[11] T. Levi, J. Tomas, N. Lewis, P. Fouillat, "Resizing methodology for CMOS analog circuits", *SPIE Microtechnologies for the New Millenium 2007*, Maspalomas, Gran Canaria, Spain, May 2007