34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

# Using digital electronic design flow to create a Genetic Design Automation tool

Y. Gendrault, M. Madec, V. Wlotzko, M. Andraud, C. Lallement, *Member, IEEE* and J. Haiech

*Abstract*—**Synthetic bio-systems become increasingly more complex and their development is lengthy and expensive. In the same way, in microelectronics, the design process of very complex circuits has benefited from many years of experience. It is now partly automated through Electronic Design Automation tools. Both areas present analogies that can be used to create a Genetic Design Automation tool inspired from EDA tools used in digital electronics. This tool would allow moving away from a totally manual design of bio-systems to assisted conception. This ambitious project is presented in this paper, with a deep focus on the tool that automatically generates models of bio-systems directly usable in electronic simulators.**

## I. Introduction

Over the past twenty years, the design of complex electronic circuits, such as microprocessors, consists of assembling standard cells, picked-up from a design kit provided by silicon factories. The design of functional microelectronic systems is possible thanks to powerful Electronic Design Automation (EDA) tools (e.g. Cadence Design Systems software suite). Today in digital electronic, the designer starts with an initial behavioral description (high-level specifications) of the targeted system and let himself be guided through the design flow step-by-step. The power of EDA tools is their ability to carry out virtual prototyping and testing, simulation, prediction and verification during the design process. Thus, the models of the standard cells are the keystone of this method.

Synthetic biology is an extension of biotechnologies that aims at designing new complex biological systems based on the assembly of biological elements decoupled from their environment [1]. Its main application fields are health care [2] and environment [3]. The design approach used in synthetic biology tends to be similar to the one used in microelectronics. At mid-term, the design of artificial biological functions is expected to be a kind of LEGO™ game, which consists of assembling a set of elementary biological standardized parts, called BioBricks [4]. This means that in synthetic biology, the existence of an equivalent Genetic Design Automation (GDA) tool would be undoubtedly an important breakthrough for the development of this field [5].

Nowadays, more than 200 tools have been developed for synthetic biology. Nevertheless, they were created for specific applications or in a given context and relate only to some specific steps of the entire design flow. Among them, GenoCAD [6] and TinkerCell [7] seem to be the most complete and the only ones that can be considered as a kind of GDA tool. In this context our goal is to build a new complete GDA tool based on existing EDA software.

The role of electronic design automation tools is presented in the first part of the paper. Then the project of complete Genetic Design Automation software, based on EDA tools is introduced. In the third part, an automated model generator, avoiding bio-engineers to have strong skills in modeling languages, is presented. A state of the art of synthetic bio-system serves, in the next part, to prove the efficiency of the model generator. As a conclusion, future plans on next versions of the software are presented.

## II. Electronic design automation tools

For the past 40 years, microelectronics is a field that has grown exponentially and digital circuits, starting with only a few thousand transistors, reach over 2 billion (e.g. Intel Xeon Nehalem-EX). This was made possible of course through technological developments (component miniaturization, manufacturing process improved, etc...), but also with the powerful design flow on which such manufacture is based.

Electronic design automation (EDA) tools help the designer to efficiently build systems. Especially for the digital parts of the system, the design flow can be automated with these tools (like Cadence software suite). The design flow for digital devices involved four main stages. The first one is the behavioral description, in which descriptions are transformed into Hardware Description Languages (HDLs) which can be directly interpreted and simulated by EDA tools. The next step corresponds to the Register Transfer Level (RTL) description. This description generates a netlist of registers and logical functions corresponding to the behavioral description and can be associated with a Gate-level description, which is a schematic view of the system. Again, EDA software automates this generation. The third stage consists in standard cell assembly also called silicon synthesis. EDA tools search automatically in libraries, provided by the microelectronics silicon foundries, the standard cells, associated to a given technology, corresponding to the RTL description. In this step,

Y. Gendrault, M. Madec and C. Lallement are with the Institut d'Electronique du Solide et des Systèmes (InESS Institute of Electronics for Solids and Systems), a research unit from the Université de Strasbourg (Unistra) and the Centre National de Recherches Scientifiques (CNRS), UMR7163, Strasbourg, France.

V. Wlotzko and M. Andraud are with Télécom Physique Strasbourg (previously known as Ecole Nationale Supérieure de Physique de Strasbourg), a Grande École of Engineering, Strasbourg, France.

J. Haiech is with Laboratoire d'Innovation Thérapeutique (LIT, Therapeutic Innovation Laboratory), a research unit from the Université de Strasbourg (Unistra) and the Centre National de Recherches Scientifiques (CNRS), UMR7163, Strasbourg, France.

Corresponding author is Yves Gendrault, (+33)368854418, e-mail: ygendrault@unistra.fr.

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

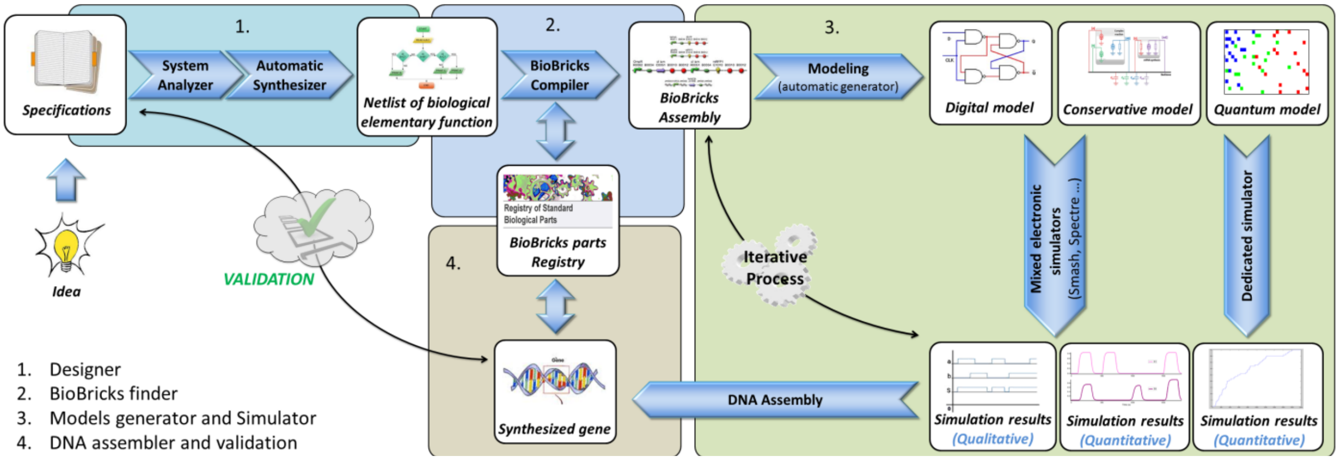
1.
2.
3.
4.
Specifications
System Analyzer
Automatic Synthesizer
Netlist of biological elementary function
BioBricks Compiler
BioBricks Assembly
Modeling (automatic generator)
Digital model
Conservative model
Quantum model
Registry of Standard Biological Parts
BioBricks parts Registry
Idea
VALIDATION
Iterative Process
Mixed electronic simulators (Smash, Spectre …)
Dedicated simulator
Synthesized gene
DNA Assembly
Simulation results (Qualitative)
Simulation results (Quantitative)
Simulation results (Quantitative)
1. Designer
2. BioBricks finder
3. Models generator and Simulator
4. DNA assembler and validation

Figure 1. Scheme of the global project of a GDA tool for synthetic biology, based on electronic languages

schematic, placement and routing are optimized to obtain the best system according to specifications provided by the designer. The last stage is to transform the standard cell assembly into assembly of transistors, called layout, which realizes the same function. This layout is send to the silicon foundries for the manufacturing after some post-processing.

## III. The GDA tool

EDA tools have proven themselves to be essential in the design of digital electronic systems and are constantly used throughout the design stages. Creation of equivalent software for the design of synthetic bio-systems, a GDA tool, is useful because many similarities exist between the electronic and synthetic biology design stages. However, some steps must be modified and reconsidered because the equipment used is no longer the silicon but mainly consists in genes. On the other hand, the interest is to keep the use of some software from EDA tools (e.g. logic synthesis). So it is necessary to provide these tools elements they can understand, as the specifications or synthesized models in HDLs.

### *A. The global suite*

The scheme Fig.1 represents the whole targeted software suite. The process starts with a biological function to design, which leads to some specifications and requirements. Since it has been demonstrated that most of the biological processes can be abstracted as digital functions [8], the first design steps could be the same as in digital microelectronics. As a consequence, HDLs and existing tools in this field, such as RTL compiler or digital synthesizer, can be used to support this part. Then, a system analyzer followed by an automatic synthesizer translates this description into a block diagram of elementary functions in Fig. 1 (part 1). During the next step (Fig. 1 (part 2)), the BioBrick compiler (eq. silicon compiler in EDA tool) finds the most relevant BioBricks to realize the system. This tool should be linked with a BioBrick database, like the Part Registry created in the context of the iGEM Competition, in which the BioBricks involved into the system can be picked up.

The third part of the software in Fig. 1 (part 3), on which the paper is focused, is a crucial part of the project. The work was aimed at translating biological mechanisms, provided by the BioBrick finder, into models coded in languages understandable by conventional electronic simulators. This is done by using analogies between electronic components and biological mechanisms [9]. Biological mechanisms are first identified and classified according to type. Standardized models are then developed for any of these reactions. In addition, these standard models can be set based on environmental data (such as pH, temperature, ...). After these models were validated, they are included in the library of models of the software suite. By this way, the software can take advantages of the power of electronic tools for simulating complex systems in a short time.

Three types of models may be used in the software. First, a behavioral model gives a qualitative all-or-none simulation and allows the bio-engineer to get a global idea of the system’s behavior. Second, a conservative model gives more accurate results. It allows evaluating the time evolution of species in a quantitative way. Third, a quantum model is used to accurately simulate the behavior of each species in the system (eq. T-CAD simulator in EDA). However it requires a dedicated simulator, which is under construction. The current version of the software includes for the time being the two first models.

Finally, the last part of the design suite (Fig. 1 (part 4)) corresponds to the back-end in EDA and consists in assembling DNA parts and validating the designed system. This part is equivalent to transistors assembly in a layout in EDA tools but generates a group of DNA parts that can be embedded in host organisms.

### *B. Automated model generator*

The large number of generated files modeling the biological mechanisms and the lack of knowledge of bio-engineers in HDLs modeling languages requires the use of a software, which automatically generate models. This tool is coded in C + + and the Qt application framework has been used to develop the Graphical User Interface (GUI). The simulator used to simulate generated models is Dolphin SMASH [10], which is one of the most complete HDLs simulation software.

The software architecture is based on two main parts: the GUI and the Engine (Fig. 2). The GUI provides the user a simple interface allowing him to define his entire system. From this window, he can communicate with the engine in order to enter the species, parameters, etc... The user builds all the reaction blocks with this interface. When the user has finished entering its system, he presses the system generation button and the Engine realizes the file generation.

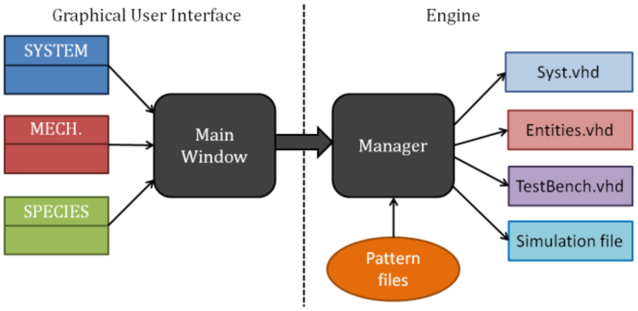

Figure 2. C++ software architecture based on two main parts: the GUI and the Engine

The simulator needs HDLs files containing the models created. To generate automatically these files associated with the system, the Engine analyses the reactions involved in the system and makes links between the different blocks corresponding to them. This Engine is the crucial part of the architecture, and a lot of work has also been done to make the model generation as automatic and reliable as possible. The Engine, linked with the GUI, fills up pattern files accordingly to the different species and reactions required by the system. These pattern files were developed to make them as generic as possible for each type of reaction.

Two versions of the software exist and can be used separately. The first version, entitled Behavioral Code Generator, allows the user to simulate a Boolean version of the reactions involved in his system. This version enables bio-engineers to have a quick overview on the system, by using very simple logic equations. It could be useful before going further into the BioBrick's conception.

The second version extends the concept previously developed to conservative models. In this version, the biological mechanisms are modeled using Ordinary Differential Equations (ODEs), which provides the user the concentrations of the species involved. These ODEs, combined with binding equations (like the Hill equation), allow modeling the mechanisms generated by the software, namely binding reaction and protein synthesis. Starting from these equations, the equivalent of electronic components has been identified, as resistors for the degradation terms, capacitors for species storage and current sources for species generation. This was already developed in previous publications [11,12] and allows obtaining conservative models.

## IV. Example of the Model Synthesizer

In this part the software efficiency is illustrated by modeling a bio-system selected from the most advanced ones in literature. The system modeled is described in Xie *et al.* [13]. This system can differentiate HeLa cancer cells from non-HeLa cells by measuring thresholds of microRNA markers. Some of these markers should be present, while others must be absent, in the HeLa cancer cell.

The first subsystem modeled with the software is shown in Fig. 3. rtTA is the synthesized activator for LacI, itself a repressor for the third gene coding for DsRed (for painting the cancer cell in red but can be replaced by a mechanism of cell destruction).

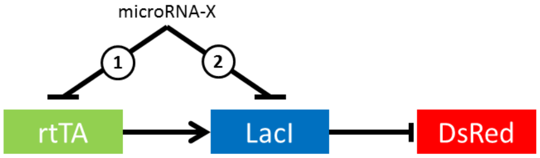

Figure 3. Subsystem illustrating the cooperativity of microRNA action on rtTA (1) and LacI (2).

The interest of this section is to highlight the cooperativity of microRNA action on the degradation of RNAs. Indeed, they allow the degradation of the RNA, but they must have effect on both rtTA and LacI RNAs, because the only effect on the LacI RNA is not enough as it can be see Fig. 4.

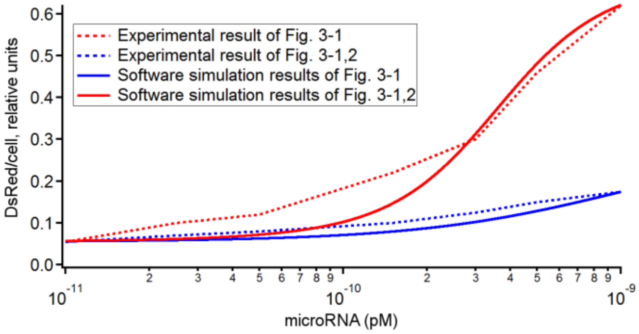

Figure 4. DC Simulation and experimental results of different Fig. 3 cases.

Fig. 4 shows the experimental results from the publication of Xie [13], in dotted line, and simulation results of the models generated by the software, in solid line, of a system where only LacI is sensitive to microRNA (Fig. 3 (2)), in blue, and an another where both rtTA and LacI are sensitive to microRNA (Fig. 3 (1) and (2)), in red. In both cases, generated models clearly fit the experimental results after adjustment of parameters. This shows that from a simple data entry in the software, it generates complicated models reaching real behavior.

However, around $10^{-10}$ pM of microRNAs, the simulated curve of the system, with the double influence of microRNAs on rtTA and LacI, presents a significant deviation with the experimental results. This is explained by the fact that the models used are based on the Hill equation, which depends only on two parameters and does not correctly model a general cooperative binding system with several thresholds. In this application, focus is made on both cases: the absence or the presence (high concentration) of miRNA. As a consequence, the observed mismatch is not detrimental for the simulation of the whole system. However integration of another type of model to have the correct behavior of binding cooperativity is under consideration.

Based on the previous subsystem, the whole system is modeled. It consists of two cells of this type, one sensitive to miR-21 and the other sensitive to miR-17 or 30a. In this system, DsRed is designed to be sensitive to three microRNAs (miR-141, miR-142 and miR-146a), to have the correct detection of a HeLa cell cancer, namely (1):

$$\begin{aligned}\text{DsRed} = \text{miR-21}\ AND\ \text{miR-17-30a}\ AND\ NOT(\text{miR-141})\\ AND\ NOT(\text{miR-142})\ AND\ NOT(\text{miR-146a})\quad \mathbf{(1)}\end{aligned}$$

Simulation results of the modeled system are presented in Fig. 5.

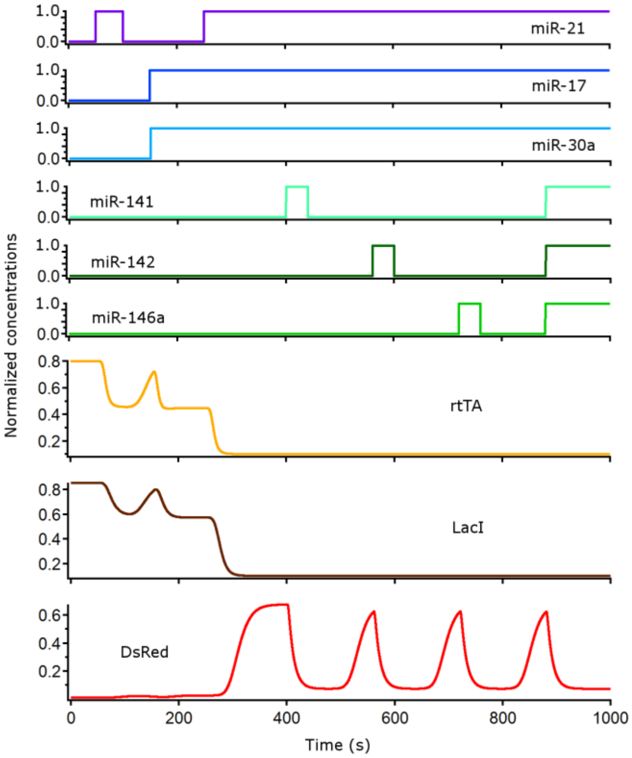

Figure 5. Transient simulation results of the whole Xie system [13]

With the presence of miR-21 or 30a-17 separately, rtTA and LacI are synthesized in a lesser concentration but always too much to activate the synthesis of DsRed. However when both are present, rtTA and LacI are completely degraded and DsRed is synthesized. The presence of miR-141, miR-142 and miR-146a, separately or together, clearly induces the degradation of DsRed.

These generated models include more than 60 files corresponding to more than 300 quantities generated. Such a system demonstrates that synthetic biology has now reached the stage where the use of software to help the designer is useful and where it begins to be difficult to model a complex system manually.

## V. Conclusion

It has been proved that electronics and biology can be combined and work together. This association is particularly important as synthetic biology is evolving towards more and more complex models and following the famous Moore's law. The software is currently able to simulate the state of the art bio-systems published, thanks to the power of electronic tools that can be very helpful for bio-engineers.

In the future, a new version of the software will include the quantum models with a dedicated simulator. Other mechanisms can be added in addition to the binding reaction and protein synthesis. Integration of communication between the multiple software parts is necessary in order to generate the different models simultaneously.

Saving and loading species and reactions created by the software in and from a file will allow the bio-engineer to get his work back between sessions and to use automation scripts.

We are investigating the possibility to code digital and conservative models in another open source HDLs [14]. This would allow integration of an open-source simulator directly in the software. An user friendly drag-and-drop interface should be developed in order to build more easily the block diagram of the system.

Finally, all the software blocks of Fig.1 will be created and integrated into the global suite meaning the release of the first and most complete GDA tool for synthetic biology.

## References

[1] R. Endy, “Foundations for engineering biology,” Nature, vol. 438, pp.449–453, 2005.

[2] V. Martin, D. Pitera, S. Withers, J. Newman, and J. Keasling, “Engineering a mevalonate pathway in escherichia coli for production of terpenoids,” Nature Biotechnol., vol. 21, p. 796802, 2003.

[3] E. Rubin, “Genomics of cellulosic biofuels,” Nature, vol. 454, p. 841845, 2008.

[4] B. Canton, A. Labno, and D. Endy, “Refinement and standardization of synthetic biological parts and devices,” Nat. Biotechnol., vol. 26, p.787793, 2008.

[5] M. W. Lux, B. W. Bramlett, D. A. Ball, and J. Peccoud, “Genetic design automation: engineering fantasy or scientific renewal?” Trends in Biotechnology, vol. 30, pp. 120–126, 2012.

[6] Genocad, http://www.genocad.org.

[7] Tinkercell, http://www.tinkercell.com.

[8] B. Kramer, C. Fischer, and M. Fussenegger, “Biologic gates enable logical transcription control in mammalian cells,” Biotechnol Bioeng., vol. 87, pp. 478–484, August 2004.

[9] M. Madec, C. Lallement, Y. Gendrault, and J. Haiech, “Design methodology for synthetic biosystems,” in Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference, 2010, pp. 621 –626.

[10] Dolphin SMASH, http://www.dolphin.fr/

[11] Y. Gendrault, M. Madec, C. Lallement, F. Pecheux, and J. Haiech, “Synthetic biology methodology and model refinement based on micro-electronic modeling tools and languages,” Biotechnology Journal, vol. 6, pp. 796–806, 2011.

[12] Y. Gendrault, M. Madec, C. Lallement, and J. Haiech, “Multi-abstraction modeling in synthetic biology,” in Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on, 2010, pp. 1–5.

[13] Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss2, and Y. Benenson, “Multi-Input RNAi-Based Logic Circuit for Identification of Specific Cancer Cells,” Science, vol. 333, pp. 1307-1311, 2011.

[14] F. Pecheux, M. Madec, and C.Lallement, “Is SystemC-ams an appropriate promoter for the modeling and simulation of bio-compatible systems ?” in IEEE International Symposium on Circuits And Systems (ISCAS), 2010, pp. 1791–1794.