Design Methodology and Modeling of Synthetic Biosystems

Morgan Madec, Yves Gendrault, Christophe Lallement, and Jacques Haiech

***Abstract*—Synthetic biology is an emerging area of biotechnology for which main applications are in the field of Health and Environment. However, it suffers from a lack of adapted CAD tools and methodology in order to fulfill efficiently and quickly the needs of these domains. In this paper, the strong relationship between circuits design in microelectronics and synthetic biology is highlighted. Most of synthesized biodevices behavior can be interpreted and modeled by BioLogic gate. As a consequence, bigger biosystems might be designed using methods and tools borrowed from microelectronics. These similarities lead to an efficient methodology, using microelectronics design flow, tools and methods, which should allow a top-down approach in synthetic biosystem design. The methodology is illustrated on the design of a biosystem (a T-flipflop), using top-down approach and HDL modeling languages. The proposed methods and their evolution prospects are discussed at the end of the paper.**

***Index Terms*—Synthetic biology, design flow, top-down approach, HDL, biological gates, biological flip-flop**

I. Introduction

SYNTHETIC biology is an emerging area of biotechnology, which aims to design and build new biological functions and systems by combining biological knowledge and engineering techniques [1]. By this way, several purposes are targeted. The first one is the improvement of our knowledge on life. Based on the assumption that biology is too complex to be integrally described and modeled, synthetic biology aims to artificially rebuild a biologic system by reproducing its functionality in order to grasp fundamental laws of life. A second goal is to develop new microorganisms or to reprogram existing ones, in order to provide them a specific function, as, for instance, synthesizing a given protein only when another one is present in the cell. In addition, synthetic biology aims to simplify the interaction with living matter: it allows driving biological actuator according to biological sensed data after a biological processing, instead of an electronic one. Obviously, synthetic biology has many applications in the field of Medicine [2], [3] but also in Environment [4], [5].

The building plane of a biological system is the genome. The genome is a network of genes, which are encoded on DNA strands. Each gene contains the necessary information needed to synthesize proteins, which cooperate to realize a biological function. One of these functions beyond others is the activation or the repression of other genes. Synthetic biology consists in designing new genes and/or small gene networks dedicated for given functions. The basic element in synthetic biology is a piece of DNA performing a given task [6]. The term BioBrick has been used to describe such standardized part of genes [7].

From a system abstraction point of view, a gene can be considered as a combinatorial logic gate, with inputs (proteins activating or repressing the expression of the gene) and output (proteins coded by the DNA strand and synthesized when the gene is expressed) [8]. The logic signal corresponds to the existence (logic-1) or the lack (logic-0) of a given protein inside the cell. As in digital electronics, interconnection between biodevices is possible as soon as the protein synthesized by the first biodevice become an activator/repressor (or regulating protein) for the second. As the behavior of BioBricks can be modeled by digital gates, methods coming from digital electronics (Karnaugh map for combinatorial circuits, Huffman methods for sequential electronics, finite states machines ...) [9], top-down design methodology, synthesis tools and languages (VHDL [10], [11], Verilog [12], [13], SystemC [14], [15] ...) should be applied in synthetic biology.

As for microelectronics, in practice, the mechanisms are more complex:

1) The expression of a gene depends on relative concentration and affinity of activator and repressor according to stochastic or continuous-time differential equations.
2) A gene can have multiple regulating proteins with different strengths.
3) Parasitic chemical reaction may appear between two active proteins or between an active and passive one.

Those considerations should be taken into account only in the low-level description. In the other hand, the system level (high-level) is an abstracted description aims to find and to validate the concept of a synthesized biosystem.

The paper describes an efficient methodology, taken from microelectronic design flow, to synthesize biosystems. Section II gives a state of the art on BioBrick synthesis methods and the associated tools. The notion of BioLogic Gate is explained in Section III. Then, the proposed methodology is exposed and applied on an example (the cell division counter). Focus is made on the low-level modeling approach (Section V). Finally, a discussion is carried out about the potential of electronic-derived design methods, constraints linked to synthetic biology, and obstacles to overcome.

II. State of the Art

Today, the development of biosystems is still an empirical process, using a bottom-up approach. First, elementary parts

M. Madec, Y. Gendrault and C. Lallement are with the Institut d'Électronique du Solide et des Systèmes (InESS), UMR 7163 (Centre National de Recherches Scientifiques / Université de Strasbourg), 23 rue du Loess, 67037 STRASBOURG CEDEX 02, France

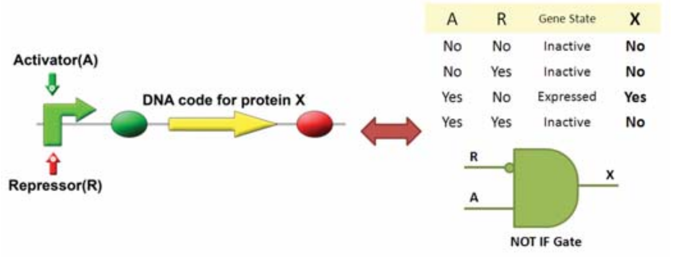
J. Haiech is with the Laboratoire d'Innovation Thérapeutiques (LIT), UMR 7200 (Centre National de Recherches Scientifiques / Université de Strasbourg), 74, route de Rhin, 67400 ILLKIRCH, France

(BioBricks) are designed using biological knowledge. After experimental validation, the majority of designed BioBricks is listed in the Registry of Standard Biological Parts [16], which is an open database managed by the MIT and containing about 3,200 parts. Bioparts are combined into a biodevice and biodevices are then combined to form a biosystem. Up to now, there is some lacks in the methodology in order to build a biosystem. Especially, a top-down approach, starting from biosystem specification and leading to the BioBricks assembly does not exist. This reached strategy remembers the one used in microelectronic for digital systems design.

Up to now, there are some tools helping the design of biosystems (e.g. COPASI [17], [18] or Virtual Cell [19], [20]). Those tools are specifically developed by and for biologists. In the other hand, it has already been demonstrated that the behavior of some BioBricks can be modeled by logic gates or electronic circuits combining passive or active components [8], [21]–[23]. This opens the way for the development of simulation software, based on electronics-dedicated existing tools. Bio-SPICE [14] is a first breakthrough on this field.

Simpson achieved a state of the art of the advances in cell modeling [23]. He points out various kinds of models for protein synthesis. He also highlights an analogy between modeling methods in electronics and in synthetic biology. This analogy has been exploited in recent approaches, using Analog and Mixed-Signal Hardware Description Languages (HDL-AMS) [24]–[27]. Many aspects of this kind of languages are very interesting for our purpose. They are suitable for modeling multi-domain, multi-abstraction, hierarchical and heterogeneous (continuous time, signal flow, event driven) systems. In addition, they are universal, standardized and human-readable by the specialist of engineering sciences as well as the biologists.

The use of such languages in synthetic biology open the way to the development of new computer assisted design (CAD) tools based on existing microelectronics CAD tools (e.g. Cadence), which have already proven its efficiency in system design [28]. Nowadays, HDLs associated with the adequate CAD tools permit to develop circuits with more than one billion of transistors (e.g. the Intel Itanium integrates about 2,000,000,000 transistors [29]) using top-down virtual prototyping methods. In such methods, the system is first described by a high level model corresponding to its specification. Then, it is refined and organized into a hierarchy in order to reach the specification of each sub-system. Each sub-system is then assembled according to the structure of the model in order to realize the system. The descending design phase is virtual and uses only the models of the components. As it is described in Section IV, some steps of the design of the virtual prototype can be automated, especially if the circuit has a digital behavior. By the same way, equivalent methods, called Functional Virtual Prototyping, are already used in the conception of heterogeneous systems in automotive (cars, motors, aircrafts...), medicine or environment [24], [30]–[32] When dealing with really complex micro total analysis systems (μTAS) or even Lab on Chips, one other possible solution to the modeling and simulation of biosystems inside some multidiscipline systems will be SystemC-AMS [33]–[36].



A	R	Gene State	X
No	No	Inactive	No
No	Yes	Inactive	No
Yes	No	Expressed	Yes
Yes	Yes	Inactive	No

Figure 1. A BioLogic NOT IF Gate

The use of HDL for the modeling of biosystems and the design of new biological functions seems to be the best way to export the experience acquired since dedades in system design for synthetic biology.

III. Biological Gates

The key point of this work is the similarity between synthesized biosystems and logic circuits. The goal of this section is to draw up this analogy.

A. The basic NOT IF gate

The basic biopart consists in a DNA strand, with a promoter, which can be activated by a protein A or repressed by a protein R, and the DNA code necessary to synthesize a protein X (Fig. 1). Considering that the repressor is stronger than the activator (which is the case for most bioparts), we obtain the Boolean function A INH $R = A \cdot \bar{R}$, that will be noticed, in the following, as a NOT IF behavior (Fig. 1).

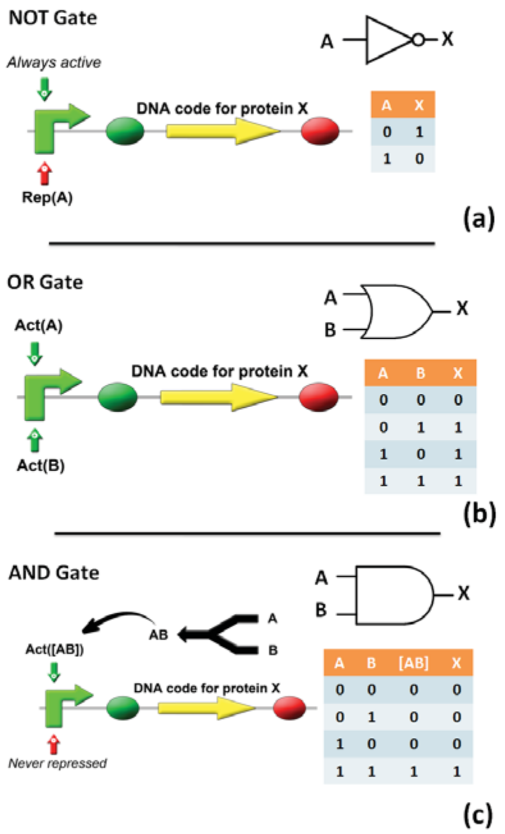
B. Standard logic gates

According to Boolean algebra properties, the NOT IF gate is a complete operators set: all the logical functions can be achieved using only a combination of some NOT IF gates. Nevertheless, as the number of interacting proteins in the cell is a big issue. To reduce this number and simplify the design of big systems, some other biological mechanisms are used in order to achieve NOT, AND and OR gates (Fig. 2).

1) NOT gate: The promoter of a gene can be constitutive (i.e. gene does not need any activator to be expressed). A gene with a constitutive promoter is always expressed except if there is a repressor in the cell. One can write $Expr =$ NOT A. By this way, a 1-gene BioLogic NOT gate can be achieved (Fig. 2a).

2) OR gate: A gene can be activated or repressed by more than one kind of proteins. Using this fact, it is possible to build a 1-gene OR gate or a 1-gene NOR gate. For example, on Fig. 2b, the gene is expressed as soon as at least one of the two activators is present ($Expr = A$ OR B). By the same way, a NOR gate is obtained by replacing the two activators by two repressors.

3) AND gate: A gene can be activated by a protein complex instead of a single protein. On Fig. 2c, the gene is expressed if the complex $[AB]$ is in the cell. The complex itself is present in the cell as soon as the two proteins involved in the complex, A and B are present in the cell. As a consequence,



A	X
0	1
1	0

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

A	B	[AB]	X
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Figure 2. NOT (a), OR (b) and AND (c) BioLogic gates. For each gate, the biopart, the digital equivalent circuit, the truth table is given

the expression of the gene is given by the following logical proposition: $Expr = A$ AND B. Thus, it is possible to build a 1-gene AND gate. By the same way, a NAND gate can be obtained if the complex $[AB]$ plays the role of a repressor instead of an activator.

C. Biodevice combining multiple bioparts: a specific NOR gate

With these basic BioBricks, it is possible to construct more sophisticated functions. As an example, we aim to build a NOR biodevice that synthesizes a green fluorescent protein (*GFP*) as long as neither *isopropyl β-D-1-thiogalactopyranoside* (*IPTG*) molecule nor *tetracycline* (*Tet*) molecule is present in the cell. The concept of such a system is described on Fig. 3. It is composed with three parts:

1) First, a gene synthesizes a specific protein, the *ZFR1*, when the molecule *IPTG* is present in the cell. The actual biological mechanism is the following. The molecule *IPTG* binds with the protein *LacI* which is a repressor for the gene. As a consequence, when *IPTG* is present, the *LacI* proteins becomes *[IPTG-LacI]* complexes. There is no more repressor in the cell and the gene can be expressed.
2) By the same way, a second gene synthesizes another protein, the *ZFR2*, when the molecule *TeT* in the cell. The mechanism is the same: *TeT* binds with a protein called (*TetR*) which is a repressor for the gene.

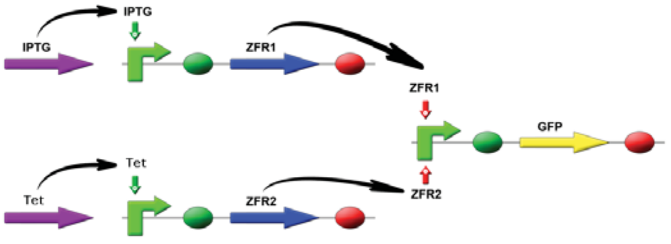


Figure 3. Example of a NOR gate designed using Zn-finger

3) The third part is a NOR gate using the mechanism described in Section III-B2: the *ZFR1* and the *ZFR2* are two repressors for a gene using a constitutive promoter and synthesizing the *GFP*. *ZFR1* and *ZFR2* are two particular proteins, called *Zinc-finger repressor* that involves interesting properties for the design of synthetic biosystems [37].

IV. Design Methodology

In this section, a method to design biosystems, based on the one used in microelectronics, is proposed (Fig. 4).

A. Front-end : from concept to part-level schematic

The front-end is a set of operations that aims to go from a concept down to a low-level structural description using BioBricks. It can be divided into 5 steps:

1) *Step 1 - High-level behavioral description*. The system to design is described at high level according to its specifications. To simplify the description and the following steps, the digital abstraction is used, as much as possible. As a consequence, HDLs (VHDL or Verilog) are the best languages to perform this step.
2) *Step 2 - Synthesis in the digital abstraction*. The synthesis consists, like in microelectronics, to step down from the behavioral description to a structural one using standard elementary gates. An automatic synthesis tool (e.g. Cadence RTL compiler tool) can be used to perform this task. Nevertheless, there are some restrictions. In particular, standard elementary gates are not the same in microelectronics and in synthetic biology: the description obtained using a microelectronic-oriented synthesis tool will be correct but may not be optimized from a biological point of view. Manual synthesis is still possible using standard methods (e.g. Karnaugh, Huffman, ...) [9]. It should be reminded that the number different proteins involved in the system is equal to the number of nodes in the electronic model. As a consequence, the complexity of the designed biosystem increases exponentially with the number of gates used to achieve the logic function.
3) *Step 3 - Schematic to model comparison*. This step consists in comparing the high-level model with the equivalent electronic schematic.
4) *Step 4 - Biobrick compilation*. The electronic schematic is interpreted and transformed in a BioBricks assembly.

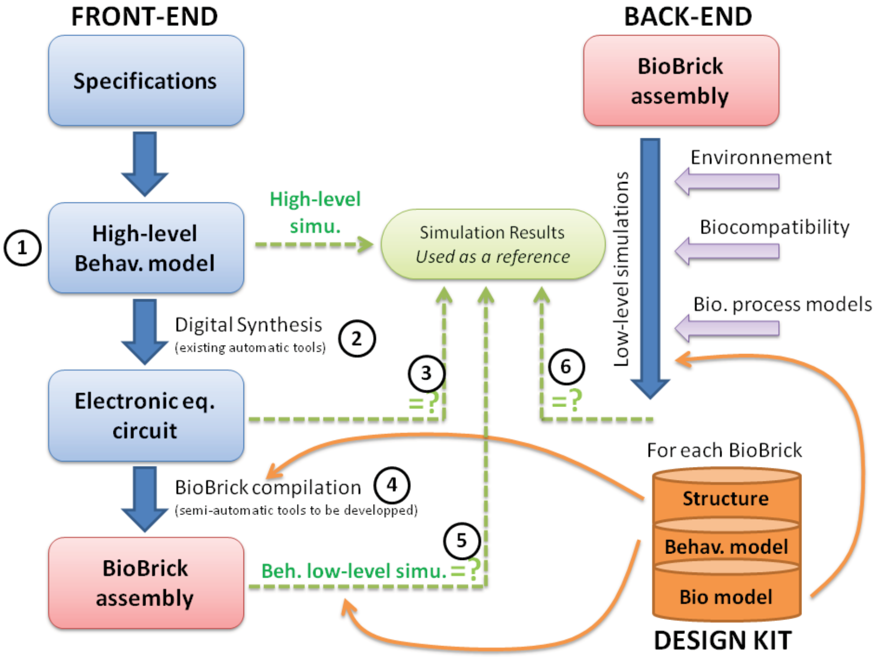


Figure 4. The proposed design flow. It is composed with six steps. The five first ones belong to the front-end whereas the last one corresponds to the back-end of the design process.

For this purpose, we have to refer to the Registry of Standard Biological Parts [16]. At this point, one of the biggest issues that are still under investigation is to find an automatic method to perform this task.

5) *Step 5 - Behavioral low-level simulation*. Every part, taken in the library, have its HDL event-driven model, including propagation times. Assembling this model, we get a low-level HDL structural model of the system. This model can be simulated and compared with the high-level one. Asynchronisms or glitches may appear at this step.

This last step ends the front-end process: the biodevice has been designed according to high-level specification and the concept has been validated through behavioral simulations.

B. Back-end : accurate simulation and system validation

The back-end (*Step 6* on Fig. 4) is a set of operation that aims to simulate the biosystem at low-level using continuous-time equations set given by biology and chemistry. The goal is to finely validate the biosystem, to predict exactly its performances (propagation time, concentrations of synthesized proteins, thresholds for regulator protein ...) and to anticipate some issues that do not appear with behavioral models. Back-end also includes experimental validations, biocompatibility tests and simulation of the biodevice in its environment. For this purpose, input block and output block have to be defined. An input block is a transport block, allowing the cell to gather inside the cell a protein that flows outside. At the opposite, an output block is a transport block allowing the cell to evacuate a protein in the environment. Such block is mandatory to design a system interacting with its environment or a multi-cell system. In addition, environmental parameters (e.g. temperature, pressure, pH ...) may alter biochemical properties of the system and should be taken into account. This point is still under investigation.

All the back-end operations require low-level, reliable and accurate models of the designed system, which is not obvious. For this purpose, a general framework has been established.

V. General Framework to Model the Biosystems

First, let us assume that the biodevice is composed with n BioBricks (B_1 to B_n), synthesizing m proteins noted $X_{i,j}$ where i is the protein number (from 1 to m) and j is the number of the incoming BioBrick. Let A_k, R_k and mX_k be, respectively, the concentration of activator, the concentration of repressor and the gene expression (quantity of mRNA) for the k-th BioBrick. The general framework that permits to model such a system is given on Fig. 5 and is composed with four kinds of block.

A. Protein synthesis

The protein synthesis model links the concentration of a given protein $X_{k,p}$ and the expression of the corresponding gene $mX_{k,p}$. In a generic model, there must be m instantiations of the protein synthesis model, one per synthesized protein. This block involves translations equations. As an example, the concentration of a synthesized protein $X_{k,p}$ as a function of the concentration of its corresponding mRNA mX_k is given by the following equation:

$$\frac{d\,[X_{k,p}]}{dt} = k_{tl,k} \cdot mX_k - d_{X_k} * [X_{k,p}] \tag{1}$$

where $k_{tl,k}$ and d_{X_k} are respectively the translation factor and the degradation coefficient for the protein $X_{k,p}$.

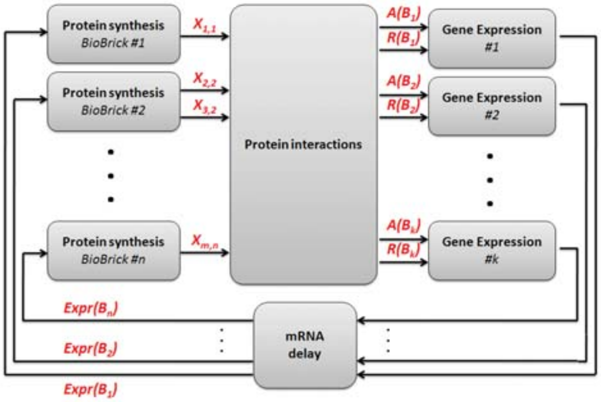


Figure 5. Framework for the low-level model

B. Gene expression

The gene expression model links, for a given BioBrick B_p, the gene expression mX_p and the concentration of its activator A_p and repressor R_p. In the full model, there must be n instantiations of such model, one per BioBrick. This block involves Hill's equations [38]. As an example, let us consider a gene, synthesizing the protein X_p, with A_p as an activator and R_p as a repressor. The concentration of mRNA coding for the protein X_p, called mX_p is given as a function of the concentration $[A_p]$ and $[R_p]$ by the following equation:

$$\frac{d\,[mX_p]}{dt} = k_{tr,p} \cdot \frac{1}{\left(1+\left(\frac{K_{A,p}}{[A_p]}\right)^{n_pA}\right)\cdot\left(1+\left(\frac{K_{R,p}}{[R_p]}\right)^{-n_pR}\right)} + k_{tr,p}\cdot a - d_{mX_p}\cdot mX_p \tag{2}$$

where $k_{tr,k}$ is the transcription factor for the mRNA strand B_p, d_{mX_p} is the degradation coefficient of the mRNA strand B_p, a is the portion of constitutive promoter, $K_{A,p}$ and $K_{R,p}$ are respectively the strength of the activator and the repressor of B_p and n_pA and n_pR are respectively the Hill coefficient of the activator and the repressor B_p.

C. Protein interaction

The protein interaction model includes all the potential interactions between active proteins $X_{k,p}$ (chemical reaction, complexation, addition, inhibition ...). For each potential interaction, a kinetic differential equation associated with a chemical balanced equation is written. For instance, let us consider two proteins, $X_{1,p}$ and $X_{2,p}$, that react in order to produce the activator A_p according to the following balanced equation:

$$X_{1,p} + X_{2,p} \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} A_p \tag{3}$$

The equation giving the concentration of A_k as a function of the concentration of $X_{1,p}$ and $X_{2,p}$is given by:

$$\frac{d\,[A_p]}{dt} = k_{on}\cdot[X_{1,p}]\cdot[X_{2,p}] - \left(k_{off}+d_{A_p}\right)[A_p] \tag{4}$$

where k_{on} and k_{off} are respectively the direct and the reverse kinetic reaction coefficients and d_{A_p} is the degradation coefficient of the protein A_p. By the same way, the remaining concentration of the protein $[X_{1,p}]$ after the reaction is given by:

$$\frac{d\,[X_{1,p}]}{dt} = k_{off}\,[A_p] - k_{on}\cdot[X_{1,p}]\cdot[X_{2,p}] - d_{X_{1,p}}\cdot[X_{1,p}] \tag{5}$$

These submodels, unmistakably the most difficult to obtain, are unique in the structural model of the biosystem and depend on the system itself.

In addition, when the system is fed back, the synthesized proteins may play the role of activator and repressor. This block also includes the relationship between the concentration of the synthesized proteins $X_{k,p}$ and the BioBrick activator A_p and R_p.

D. Delays

Delays appear along the whole process. There are two kinds of delay.

First, the chemical interactions (inhibition, gene activation, gene repression, mRNA and protein synthesis ...) are not immediate. There is a transient phase leading to a delay between an alteration of the species concentration and its effect on the cell. As we choose to represent each mechanism by a time-dependant differential equation, those delays are, by nature, included in the model.

Second, the molecules have to travel inside the cell in order to interact. This leads to a spatial-dependant delay which may depends on many parameters. In our model, at the first order, we consider that the delay is constant for all species in the cell. As a consequence, the spatial delay can be modeled by a single delay integrated in the mX_k feedback loop.

VI. Validation on Examples

The concepts presented in this paper are illustrated through two examples.

A. A basic biodevice : a NOR gate

First, we deal with the NOR gate discussed in Section III-C. The front-end for this device is obvious and does not merit discussion. Focus is made on the low-level modeling. Let us assume that there is no interaction between the involved proteins. According to the framework described in Section V, the behavior of the biosystem can be modeled according to six equations:

- 3 gene expression equations (Eq. 2) modeling respectively the transcription of the *ZFR1*'s mRNA driven by *IPTG*, the transcription of the *ZFR2*'s mRNA driven by *TeT* and the transcription of the *GFP*'s mRNA driven by *ZFR1* and *ZFR2*.
- 3 protein synthesis equations (Eq. 1) modeling respectively the translation of the *ZFR1*'s mRNA, the translation of the *ZFR2*'s mRNA and the translation of the *GFP*'s mRNA.

The model of this NOR gate has been encoded in VHDL-AMS (Listing 1) and simulated with Dolphin SMASH. Simulation results are given on Fig 6. The NOR behavior is confirmed: *GFP* is synthesized only when *IPTG* and *TeT* are not present.

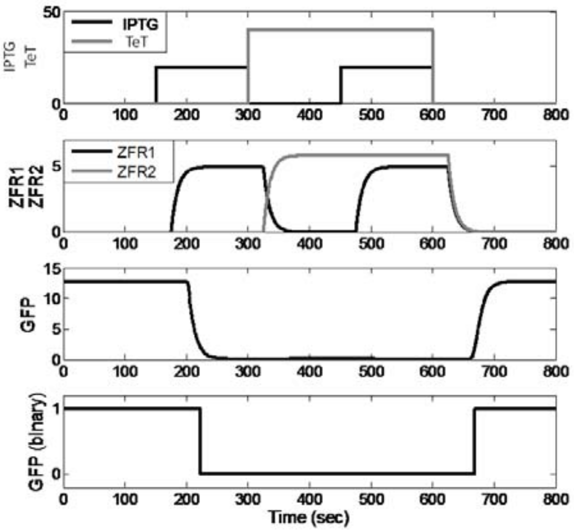


Figure 6. Simulation results. The first figure corresponds to the stimuli (*IPTG* and *TeT*). Second one and third one give the concentration of synthesized protein *ZFR1*, *ZFR2* and *GFP*. The last figure is a binary representation of *GFP* for which the threshold has been arbitrary fixed at 5.

```
ENTITY NOR_Gate IS
GENERIC (
   ktr1 , ktr2 , ktr3  :  REAL := 1.0;             −− transcription  coefficients
   ktl1 , ktl2 , ktl3  :  REAL := 1.0;             −− translation  coefficients
   dZFR1,dZFR2,dGFP : REAL := 0.1;         −− protein degadation  coefficients
   dmZFR1,dmZFR2,dmGFP : REAL := 0.1; −− mRNA degradation coefficients
   KA1, KA2, KR31, KR32 : REAL := 2.0; −− activator and repressor  strength
   n1,  n2,  n3  :  REAL := 3.0);                  −− Hill  coefficients
PORT (
   IPTG, TeT : IN  QUANTITY;
   GFP          : OUT QUANTITY);
END ENTITY;

ARCHITECTURE Arch1 OF NOR_Gate IS
   QUANTITY mZFR1, mZFR2, mGFP : REAL := 0.0;
   QUANTITY ZFR1, ZFR2 : REAL := 0.0;
BEGIN

   −− Transcription equations
   mZFR1'dot == ktr1 / (1.0 + (KA1 / IPTG)∗∗n1 ) − dmZFR1 ∗ mZFR1;
   mZFR2'dot == ktr2 / (1.0 + (KA2 / TeT)∗∗n2 ) − dmZFR2 ∗ mZFR2;
   mGFP'dot == ktr3 / (1.0 + (ZFR1 / KR31)∗∗n3) / (1.0 +
                        (ZFR2 / KR32)∗∗n3)) − dmGFP ∗ mGFP;

   −− Translation equations ( including  mRNA delay)
   ZFR1 == ktl1 ∗ mZFR1'delayed(10.0) − dZFR1 ∗ ZFR1;
   ZFR2 == ktl2 ∗ mZFR2'delayed(10.0) − dZFR2 ∗ ZFR2;
   GFP == ktl3 ∗ mGFP'delayed(10.0) − dGFP ∗ GFP;

   −− Internal chemical  reaction  :  no  equations

END ARCHITECTURE;
```

Listing 1. VHDL-AMS model of the BioLogic NOR Gate

```
ENTITY T_flipflop IS
GENERIC (
   Tswitch  :  TIME := 5 min);
PORT (
   APC      :  IN  BIT;
   GFP,YFP : OUT BIT);
END ENTITY;

ARCHITECTURE Behavior OF T_flipflop IS
   SIGNAL cell_state : BIT := '0';
BEGIN
   PROCESS (APC) BEGIN
      IF (APC'event AND APC='1') THEN
            cell_state   <= NOT cell_state AFTER Tswitch;
      END IF;
   END PROCESS;

         GFP <= '1' WHEN cell_state = '0' ELSE '0';
         YFP <= '1' WHEN cell_state = '1' ELSE '0';
END ARCHITECTURE;
```

Listing 2. VHDL high-level description of the BioLogic T-flipflop

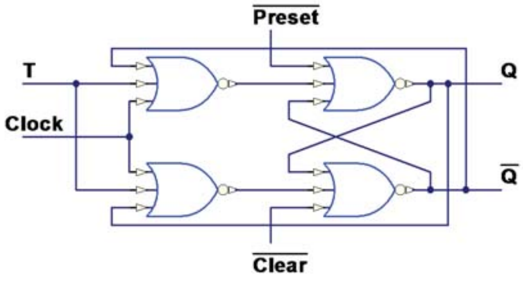


Figure 7. Schematic of an electronic T-flipflop using four 3-input NOR gates

B. A complex biodevice : APC-driven a T-flipflop

The second example is the reprogramming of a yeast in order to achieve a T-flipflop driven by the *anaphase-promoting complex* (*APC*), a protein used in the cell cycle regulation. The work presented in this paper won the bronze medal at iGEM (a inter-university international challenge organized by the MIT) in 2008 [39], [40].

1) High-level description: At each cell cycle, the cell switches between a state '0' where it synthesizes a green fluorescent protein (*GFP*) and a state '1' where it synthesizes a yellow fluorescent protein (*YFP*). From a high-level point of view, the biosystem is an oscillator and can be described, in VHDL, according to a behavioral description given in Listing 2. The parameter `Tswitch` is used in order to describe the delay between the apparition of the *APC* and the end of the switching process.

2) Gate-level view: The second step consists in the logic synthesis, starting from the behavioral VHDL description and leading to a gate-level schematic. Up to now, no automatic synthesizer exists. Nevertheless, as this device is a standard in digital electronics, the drawing of the gate-level view is straightforward.

The gate-level circuit is built upon a digital T-flipflop schematic (Fig. 7). Generally, in digital electronics, 2-input NOR gates are used (in our system, *T* is always active and there is no *Set/Reset* mechanism). The NOR gates describes in Section III-C could have been used. The realization of a T-flipflop using such NOR gates would have required 5 genes (one per NOR gate and one buffer to convert *APC* in a Zn-Finger) and 7 different Zn-Finger proteins.

By analysing the circuit with a biological point of view, a simplified schematic can be drawn (Fig. 8). Both schematics are equivalent according to Boolean algebra rules, but the biological implementation of the circuit in Fig. 8 requires only 2 genes and 6 different proteins.

The efficiency of the designed circuit can be easily and quickly checked using a digital event-driven simulation (solid lines in the graph on Fig. 11).

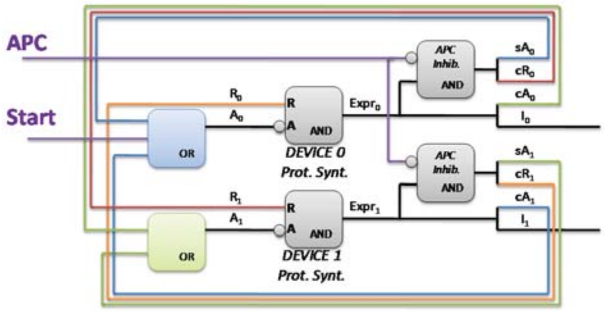


Figure 8. Scheme of an BioLogic T-flipflop using OR and NOT IF gates

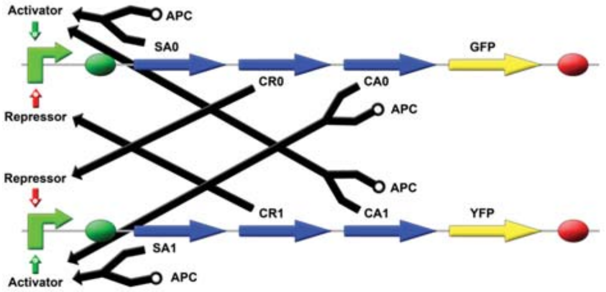


Figure 9. BioBrick assembly leading to the T-flipflop behavior. The system is composed with two genes that are alternatively expressed or not.

3) Biobrick-level view: The last step of the front-end design flow consists in transforming the BioLogic gate (Fig. 8) schematic into a BioBrick assembly. The schematic involves OR and NOT IF gates. The realization of the OR gate uses the principle described in Section III-B2. For instance, for the first OR gate, the protein called *Start*, *sA0* and *cA1* plays the role of activators.

The two first NOT IF gates (marked *DEVICE 0* and *DEVICE 1* on Fig. 8) are also conform to the description given in III-A, using a protein as activator and a protein as repressor.

The second NOT IF gate, involving *APC*, uses a tag-mechanism. The protein that should be inhibited when *APC* is high, are tagged. As a consequence, they are degraded by the *APC* molecules and become inactive. From a biological point of view, the use of such mechanism is very interesting because it permits to realize a logic operation without using a gene. Finally, the BioBrick involves only two *devices* (Fig. 9).

Let us analyze the system from a mechanical point of view. At the beginning, an injection of *Start* protein leads to an activation of the *DEVICE 0* that synthesizes *GFP* (system is in the 0-state), *sA0* which is a self-activator maintaining the expression of *DEVICE 0* and two proteins, *cA0* and *cR0*, which are respectively a cross-activator and a cross-repressor for *DEVICE 1*. As the repressor is stronger than the activator, *DEVICE 1* is never expressed and the system is in a steady state. When *APC* is synthesized by the cell, *sA0* and *cR0* disappear. Thus, the activity of *DEVICE 0* decreases progressively whereas the degradation of repressor lead to an increase of *DEVICE 1* expression. When *APC* disappears, the *DEVICE 1* is expressed whereas the *DEVICE 0* is inhibited. As the *DEVICE 0*, the *DEVICE 1* synthesizes the *YFP*, a self-activator (*sA1*), a cross-activator (*cA1*) and a cross-repressor (*cR1*). The coexistence of these three last proteins in the cell leads to a second steady state. The biosystem switches from one steady state to the other according the previous mechanism as soon as *APC* is synthesized in the cell.

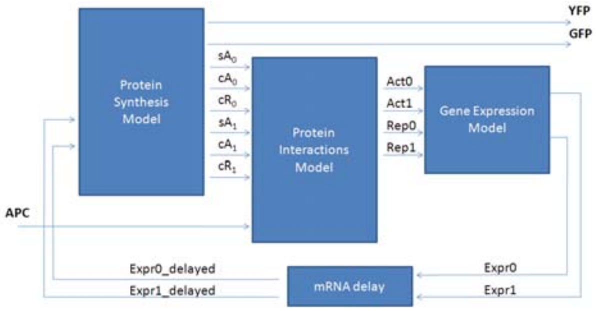


Figure 10. Block diagram for the low-level modeling of the T-flipflop BioBrick assembly.

4) Back-end : low-level model: The last step of this work consists in building a low-level model of the system and verifying that the T-flipflop behavior using biological equations. The low-level model is obtained according to the framework presented in Section V. The block diagram of the model is given in Fig. 10.

The *Gene Expression model* and the *Protein synthesis model* correspond exactly to the description given in Section V-A and V-B. The differential equations are coded directly in VHDL-AMS. The *mRNA delay model* is a simple delay obtained with the VHDL-AMS attribute `'delayed()`. Finally, the *Protein Interaction model* contains some equations describing the potential interaction between activator and repressor as well as the degradation of the species in presence of *APC* .

First, the degradation of *sA0*, *sA1*, *cR0* and *cR1* driven by *APC* are described with the following equation.

$$\frac{d[\widetilde{X}]}{dt} = [X] - (1 + k_{APC} * [APC]) \cdot [X] \tag{6}$$

where $[X]$ is the quantity of protein synthesized by the gene, $[\widetilde{X}]$ is the quantity of protein after degradation and k_{APC} is the degradation factor. In the model, Eq. 6 is implemented four times, one per APC-targeted protein.

The static resolution of Eq. 6 leads to the following equality:

$$\left[\widetilde{X}\right] = [X] \cdot \frac{1}{1 + k_{APC} * [APC]} \tag{7}$$

Eq. 7 confirms the expected behavior: when $[APC] = 0$, there is no degradation and $[\widetilde{X}] = [X]$ whereas when $[APC]$ increases, the protein is degraded and $[\widetilde{X}]$ tends to 0.

Second, activator and repressor for each gene are in competition in the cell. As a consequence, the *Protein Interaction model* have to compute the effective quantity of activator and repressor for each gene (*A0*, *A1*, *R0* and *R1*) as a function of *sA0*, *sA1*, *cR0*, *cR1*, *cA0* and *cA1*. The equation implemented in the model for the activator and for the repressor are the following:

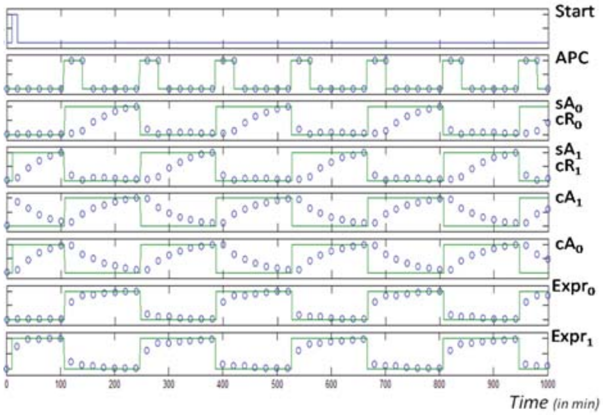


Figure 11. Simulation results. Lines correspond to the event-driven simulation of the digital model whereas dots corresponds to the continuous-time simulation of the low-level model. Transient variation of the involved protein are given. The value of the concentration of each species has been normalized in order to vary between 0 and 1.

$$[A_x] = \frac{1}{sA_x + cA_{1-x} + cR_{1-x}} \cdot \Big((sA_x + cA_{1-x})^2 + 2 \cdot (sA_x + cA_{1-x}) \cdot cR_{1-x} \cdot c_{A,R}\,(1 - \eta_{A,R}) \Big) \tag{8}$$

and

$$[R_x] = \frac{cR_{1-x}^2 + 2 \cdot (sA_x + cA_{1-x}) \cdot cR_{1-x} \cdot c_{A,R} \cdot \frac{\eta_{A,R}}{K_m}}{sA_x + cA_{1-x} + cR_{1-x}} \tag{9}$$

where $c_{A,R}$ is the affinity constant between activator and repressor, $\eta_{A,R}$ is the ratio of $[AR]$ complex that can act as a repressor and K_m ratio of strength between activator and repressor. The biological details about this equation are not given in this paper.

In the low-level model, the use of a *Start protein* is not necessary because the quantities are initialized in order to by in one of the two steady states.

5) Results: The model is implemented in VHDL-AMS and simulated with Advance-MS. The simulation results are given in Fig. 11. The solid lines correspond to the event-driven model whereas the dots correspond to the normalized continuous-time simulation. Both simulations are equivalent and represent a 140-min oscillating system.

By a manual back-annotation mechanism, the propagation time in the event-driven VHDL model have been adjusted in order to correspond to the settling time measured with the continuous-time model.

VII. Discussion and Conclusion

The paper proposes a design flow allowing rational design of biosystems from identified and standardized BioBricks. The methodology is inspired from microelectronic design methods. It is separated into two main steps: a top-down design flow (starting with a behavioral HDL model of the system and providing, after some manual or automatic steps, a BioBricks interconnection scheme), and a validation process (using continuous-time models of BioBricks and a given framework, environmental interaction, ...).

Although, some steps of the design flow have to be improved, formalized and/or automated (BioLogic synthesizer, standardization of BioBrick, improvement of the continuous-time model ...), the proposed methodology already allows designing effectively small biosystems. The example of the T flip-flop given in this paper illustrates the power of our approach.

Synthetic biology is still an emerging domain. Design on biosystems can take advantages of 60 years experience in design techniques in microelectronics (methodology, tool and languages ...). The main outlines drawn in this paper is a first approach that highlights this point.

References

[1] R. Endy, “Foundations for engineering biology,” *Nature*, vol. 438, no. 24, pp. 449–453, 2005.

[2] F. Reza, K. Chandran, M. Feltz, A. Heinz, E. Josephs, P, O’Brien, B. V. Dyke, H. Chung, S. Indurkhya, N. Lakhani, J. Lee, S. Lin, N. Tang, T. LaBean, L. You, F. Yuan, and J. Tian, “Engineering novel synthetic biological system,” *IET Synthetic Biology*, vol. 1, pp. 48–52, 2007.

[3] Z. Ying, C. Zhou, and O. Oglesby, “Large-scale drug function prediction by integrating qid d2 in biospice,” in *Computational Systems Bioinformatics Conference*, pp. 118–122, 2002.

[4] J. Aleksic, F. Bizzari, Y. Cai, B. Davidson, K. de Mora, S. Ivakhno, S. Seshasayee, J. Nicholson, J. Wilson, A. Elfick, C. French, L. Kozma-Bognar, H. Ma, and A. Millar, “Development of a novel biosensor for the detection of arsenic in drinking water,” *IET Synthetic biology*, vol. 1, pp. 87–90, 2007.

[5] G. van den Bogaart, V. Krasnikov, and D. Poolman, “Dual-color fluorescence-burst analysis to probe protein efflux through the mechanosensitive channel mscl,” *Biophysics Journal*, vol. 94, no. 4, pp. 1233–1240, 2007.

[6] T. F. Knight, “Idempotent vector design for standard assembly of biobricks,” *MIT Synthetic Biology Working Group*.

[7] The BioBrick Fundation website. *http://bbf.openwetware.org*.

[8] B. Kramer, C. Fischer, and M. Fussenegger, “Biologic gates enable logical transcription control in mammalian cells,” *Biotechnology and Bioengineering*, vol. 87, no. 4, pp. 478–484, 2004.

[9] J. R. Gibson, *Electronic Logic Circuits*. ed. Butterworth-Heinemann Ltd, 1992.

[10] EDA Industry Working Groups. *http://www.vhdl.org*.

[11] P. Ashenden, *The Designer Guide to VHDL*. ed. Morgan Jaufmann Publisher Inc., 2008.

[12] Verilog dot com website. *http://www.verilog.com*.

[13] P. Ashenden, *Digital Design Verilog: An Embedded Systems Approach Using Verilog*. ed. Morgan Jaufmann Publisher Inc., 2007.

[14] System C website *http://www.systemC.org*.

[15] T. Grotker, S. Liao, G. Martin, and S. Swan, *System Design With SystemC*. ed. Kluwer Academic Publishers, 2002.

[16] Register of standard biological parts. *http://www.partsregistry.org*.

[17] COPASI homepage. *http://www.copasi.org*.

[18] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer, “Copasi: a complex pathway simulator,” *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 2006.

[19] L. Loew and J. Schaff, “The virtual cell: a software environment for computational cell biology,” *TRENDS in Biotechnology*, vol. 19, no. 10, pp. 401–407, 2001.

[20] The Virtual Cell homepage. *http://www.ibiblio.org/virtualcell/*.

[21] D. Gerber and M. Fussenegger, “Mammalian synthetic biology: Engineering of sophisticated gene network,” *Journal of Biotechnology*, vol. 130, pp. 329–345, 2007.

[22] R. Weiss, G. Homsy, and T. K. Jr., “Toward in vivo digital circuits,” in *DIMACS Workshop on evolution as computation towards in vivo digital circuits*, 2005.

[23] M. Simpson, C. Cox, G. Peterson, and G. Sayler, “Engineering in the biological substrate: information processing in genetics circuits,” *Proceedings of the IEEE*, vol. 92, no. 5, pp. 848–863, 2004.

[24] F. Pecheux, C. Lallement, and A. Vachoux, “Vhdl-ams and verilog-ams as alternative hardware description languages for efficient modeling of multidiscipline systems,” *IEEE Transactions on Integrated Circuits and Systems*, vol. 24, no. 2, pp. 204–225, 2005.

[25] M. Madec, C. Lallement, K. Karstens, S. Dittman, M. Gersbacher, R. Sorg, M. Wild, M. Muller, P. Bourguine, M. Donzeau, and J. Haiech, “Synthetic biology and microelectronics: a similar design flow,” in *Proc. of Joint 7th International IEEE NEWCAS-TAISA 2009 Conference, Toulouse (France), June 28 July 1*, 2009.

[26] R. Krishnan and C. Purdy, “Circuit development using biological components: Principles, models and experimental feasibility source,” *Analog Integrated Circuits and Signal Processing*, vol. 56, no. 1-2, pp. 153–161, 2008.

[27] M. Madec, Y. Gendrault, C. Lallement, and J. Haiech, “Design methodology for synthetic biosystems,” in *18th IEEE International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES), Wroclaw (Poland)*, 2010.

[28] CADENCE website. *http://www.cadence.com*.

[29] Intel website. *http://www.intel.com*.

[30] Y. Hervé, “Virtual prototyping in vhdl-ams,” in *IEEE International Conference on Industrial Technology*, 2003.

[31] Y. Hervé and P. Desgreys, “Functional virtual prototyping design flow and vhdl-ams,” in *Forum on Specification and Design Languages*, 2006.

[32] P. Poure and S. Weber, “Vhdl-ams modeling for virtual prototyping of very large scale integration architecture for direct torque control of an ac machine,” in *32nd IEEE Annual Conference on Inductrial Electronics*, 2006.

[33] A. Vachoux, C. Grimm, and K. Einwich, “Towards analog and mixed-signal soc design with systemc-ams,” in *IEEE International Workshop on Electronic Design, Test and Applications (DELTA)*, 2004.

[34] A. Vachoux, C. Grimm, and K. Einwich, “Analog and mixed signal modelling with systemc-ams,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2003.

[35] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, “Modeling and refining heterogeneous systems with systemc-ams: Application to wsn,” in *Proceedings IEEE Date 2008 Conference, Munich*, 2008.

[36] 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), Paris (France)*, 2010.

[37] T. Sera, “Zinc-finger-based articifial transcription factors and their application,” *Advanced Drug Delivery Reviews*, vol. 61, pp. 513–526, 2009.

[38] M. Terzer, M. Jovanovic, A. Choutko, O. Nikolayeva, A. Korn, D. Brockhoff, F. Zurcher, M. Friedmann, R. Schutz, E. Zitzler, J. Stelling, and S. Panke, “Design of a biological half adder,” *IET Synthetic Biology*, vol. 1, pp. 53–58, 2007.

[39] International Genetically Engineered Machine (iGEM) competition homepage. *http://igem.org*.

[40] ESBS team homepage for iGEM 2008. *http://2008.igem.org/Team:ESBS-Strasbourg*.

Morgan Madec was born in 1980. He received the M.S. and PhD degrees in microelectronics from the Université Louis Pasteur (ULP), Strasbourg, France, in 2003 and 2006 respectively. From 2003 to 2006, he was with the Laboratoire de Physique et Application des Semi-Conducteurs (PHASE), ULP-Centre National de Recherches Scientifiques (CNRS), Strasbourg, where he prepared a PhD thesis on the design, the simulation and the characterization of optical processors in order to speed up image reconstruction in the medical field. He is currently a Associate Professor with the Institut d’Électronique du Solide et des Systèmes, Université de Strasbourg (UdS), Strasbourg and teaches electronics in the Ecole Nationale Supérieure de Physique de Strasbourg, UdS, Illkirch, France. His research interests include compact modeling of integrated microsensors (Hall-effect sensor, photodiode ...). Since 2008, he collaborates with a team of the Laboratoire d’Innovation Thérapeutique. The aim of this work is to put the experience in microelectronics system design to good use in synthetic biology.

Yves Gendrault obtained his Bachelor’s degree of the Université de Strasbourg (UdS) in Electronics, Signal and Automatics (ESA) in 2008 and the M.S. degree of the UdS in Micro-and Nano-Electronics (MNE) in 2010. In february 2010, he start an internship work in the Institut d’Électronique du Solide et des Systèmes, UdS-Centre National de Recherches Scientifiques (CNRS), Strasbourg, France. The goal of this work was to find a way to adapt of microelectronics design tools and methods for the Synthetic Biology. Since september 2010, he pursues his investigations in this field as a PhD student.

Christophe Lallement received the M.S. degree in engineering from the Université des Sciences de Nancy I, Nancy, France, and the Ph.D. degree in engineering from the École Nationale Supérieure des Télécommunications, Paris, France. From November 1994 to September 1997, he was a Postdoctoral Research Scientist with the Laboratory of Electronics of the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, working on the characterization and modeling of the metal-oxide-semiconductor field effect transistor (MOSFET) in the development team of the Enz-Krummenacher-Vittoz MOSFET model. In September 1997, he was an Associate Professor with the Université de Strasbourg (UdS), Strasbourg, France, and the Laboratoire de Physique et Application des Semi-Conducteurs, Centre National de la Recherche Scientifique (CNRS). Since September 2003, he has been a Professor with the École Nationale Supérieure de Physique de Strasbourg, Illkirch, France. He is currently with the Institut d’Électronique du Solide et des Systèmes (InESS), UdS-CNRS, working on the study and the modeling of advanced devices, very-high-speed integrated-circuit hardware description language analog and mixed-signal systems, and biosynthetic systems. He is the responsible for the group ’Integrated Instrumental Systems’ at InESS.

Jacques Haiech got a M.D. degree in Mathematics and computer science in 1975 and then a M.D. and a PhD degree in Biochemistry in 1978 dealing with cell signaling (focusing on calcium signal in muscle cells). After his PhD, he has been working part time successively at NCI in Bethesda, then at Vanderbilt University and finally at Northwestern University (Chicago) while being research director at Centre National de la Recherche Scientifique (CNRS) in France, studiyng cellular calcium signals before joining the Strasbourg University as a full professor in 1997. He has founded the first synthetic biology option in an engineering school in France in 2008 along with the first participation of the ESBS-Team to the iGEM competition. Since then, he has developed collaboration with the Strasbourg engineering school of physics on Design Methodology and Modeling of Synthetic Biosystems. He is now working in integrating concepts of synthetic biology in personalized medicine.