Supplementary Information: An automated design framework for multicellular recombinase logic.

Sarah Guiziou[1], Federico Ulliana[2], Violaine Moreau[1], Michel Leclere[2], and Jerome Bonnet*[1]

[1]Centre de Biochimie Structurale, INSERM U1054, CNRS UMR5048, University of Montpellier, France.
[2]Laboratoire d'Informatique, de Robotique et de Microelectronique de Montpellier (LIRMM). CNRS UMR 5506, University of Montpellier, France.
*To whom correspondence should be addressed: jerome.bonnet@inserm.fr

These supplementary materials contain:
- -Supplementary Figures S1 to S3.
- -Supplementary Tables S1.

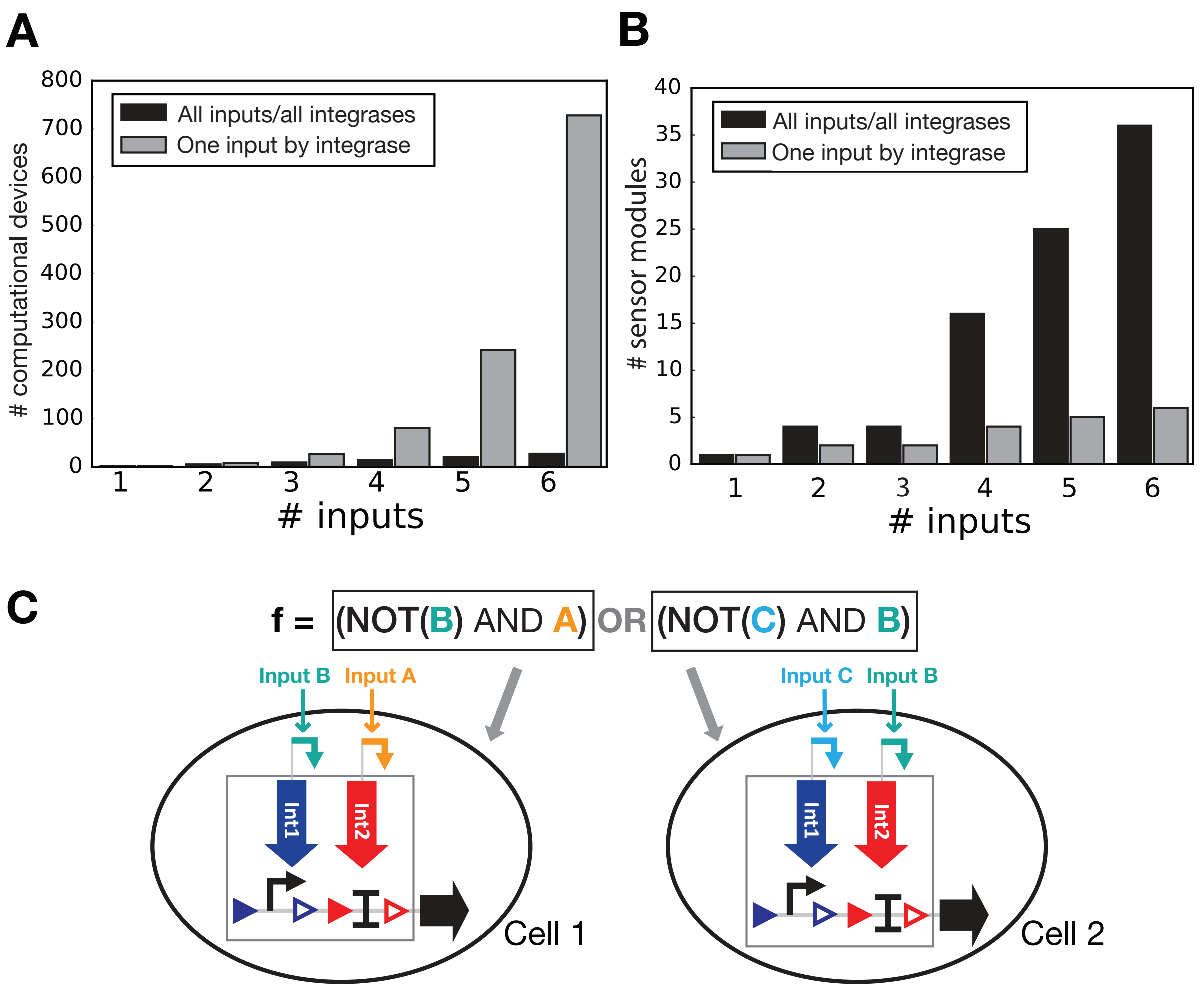

Figure S1: **Reduction of the number of Boolean logic devices by connecting all inputs to all integrases. (A)** Reduction of the number of computational devices needed by connection of all inputs to all integrases. The bar graph represents the number of standard computational devices needed to implement a function responding to a specific number of inputs, with the black bars for connection of all inputs to all integrases and the grey bars for connection of one input to one integrase (see methods for equation). **(B)** Number of sensor modules needed using all-input/all-integrase design or one input by integrase design. If only one device per symmetric function is implemented, all combinations of inputs with integrases have to be built to implement all logic sub-functions. The number of sensor modules with this design strategy is higher than in the one input by integrase design. The bar graph represents the number of sensor modules needed in function of the input number, for all-input/all-integrase design (black bars) and one input by integrase design (grey bars). By comparing A and B, it is clear that the total number of component needed is greatly in favor of the all-input/all-integrase design. **C** - Example of Boolean logic implementation based on two cells using the same computational devices and different input-integrase connections.

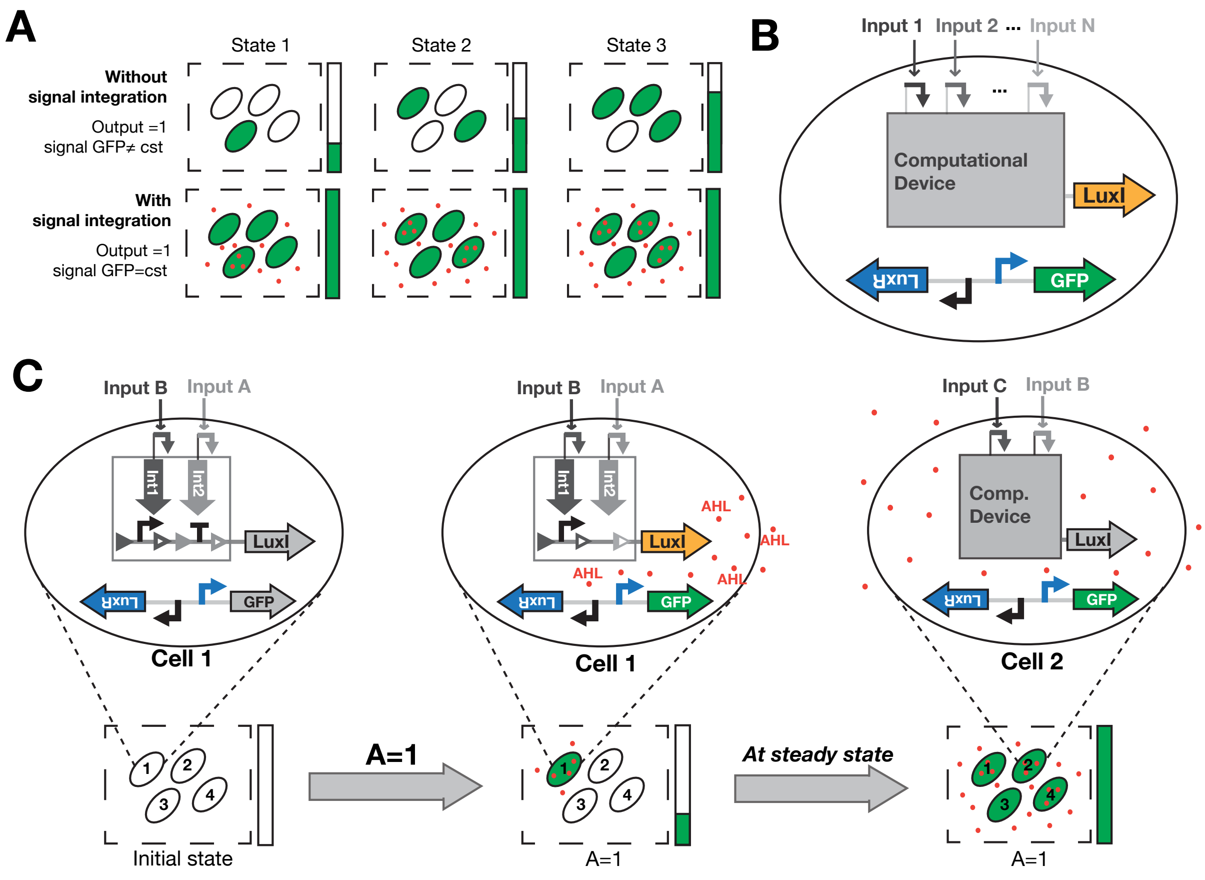

Figure S2: **Use of cell-cell communication to obtain a constant output signal between states.** **A** - The integration of the output signal is required to obtain a uniform output in all ON states. In our multicellular design, the output is considered equal to one if at least one cellular computing unit is ON. Therefore, the expression level of the output gene will be different if one or several units are ON. For applications that require a constant output level, integration of the output signal might be performed using cell-cell communication. If one of the strains is ON, it produces an AHL molecule that is detected by the other strains, which subsequently turn ON such that in all ON states of the program the output level is constant. **B** - Implementation of cell-cell communication to integrate output signals. The output gene of the computational device is a gene producing an AHL molecule (for example LuxI), and the output gene (here GFP) is connected to a promoter inducible by AHL. **C** - Example of the behavior of a cellular computing unit with a signal integration system. In the initial state for this specific strain, the output gene of the computing device (LuxI) is OFF as for all other strains. Then, with the presence of the input A, the terminator is excised, LuxI is expressed, and AHL is produced. GFP will be expressed in this strain, and by diffusion of AHL all strains will produce GFP.

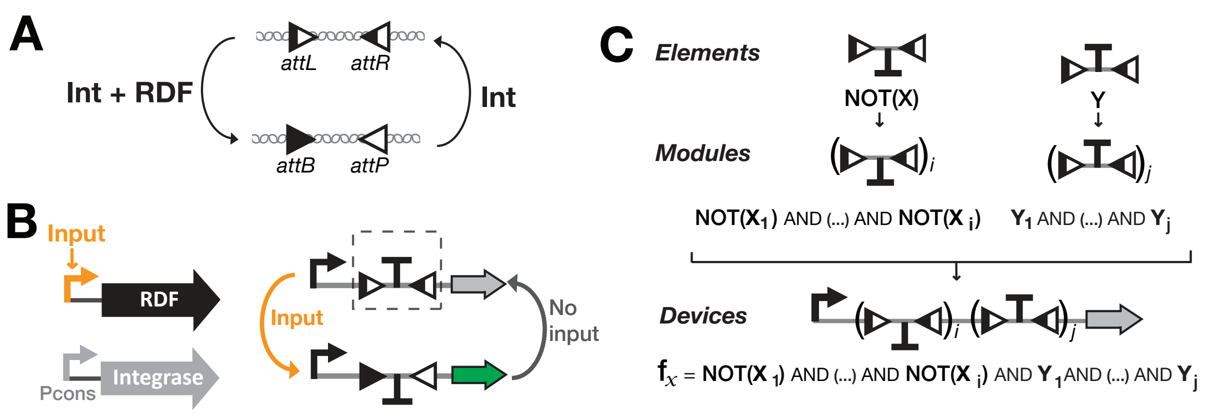

Figure S3: **Hierarchical composition framework for synchronous Boolean logic using integrases and recombination directionality factors (RDFs).** **A** - Reversible inversion of DNA using integrase coupled with RDF. Integrase alone specifically targets *attB* and *attP* sites and does not operate on *attL* and *attR* sites. When sites are oriented in the opposite direction, DNA between sites is inverted and *attL* and *attR* sites are formed. With the additional use of a RDF, the integrase targets specifically *attL* and *attR* sites and inverts DNA between these sites, reverting to *attB* and *attP*. Therefore, using a RDF enables the implementation of reversible integrase-based DNA switches[14]. **B** - To obtain a synchronous IDENTITY function, the integrase sites *attL* and *attR* are placed in inverted orientation around an asymmetric terminator. The terminator blocks the flow of RNA polymerase, and the output gene is not expressed. The integrase is constitutively expressed in all states. When the input is present, RDF is expressed and the terminator inverted. As the terminator is asymmetric, the output gene is expressed[10,30]. **C** - Hierarchical composition of synchronous elements. NOT- and ID-elements are composed with *attL* and *attR* sites in inversion mode flanking an asymmetric transcriptional terminator. For the NOT-element, the terminator is in the OFF position and for the ID-element in the ON position. ID- and NOT-modules are both composed in series between the promoter and the output gene and compute, respectively, the conjunction of IDENTITY functions and NOT functions. The device is then expandable by addition of elements in series to all logic functions based on the conjunction of NOT and IDENTITY functions.

# Strains	# 3-input functions	# 4-input functions
1	26	80
2	130	1804
3	88	13472
4	10	28904
5	–	17032
6	–	3704
7	–	512
8	–	26

Table 1: **Proportion of Boolean functions implementable with a specific number of strains for 3 and 4 inputs.** This table was obtained by systematic generation of the biological design of all 3 and 4-input Boolean functions using our python software.