

Enzyme-based computing systems: The half-adder and half-subtractor

Have we ever stopped to think what would happen if the silicon based computers that we have come to rely on reach the peak of their development? Many scientists who work in the field of unconventional computing have sought to answer that question by using many different medias, an example of which are the biochemists. Using biomolecular species such as DNA/RNA, oligopeptides, proteins, enzymes and even whole biological cells, biochemists are able to design systems with specific signal controlled properties that can mimic computational processes. Although progress in the field is still far away from re-creating a computer, the ability to create a “toolbox” of computer parts is still equally important.

Of the computer parts that have been re-created, the creation of an enzyme based half-adder and half-subtractor in a fluidic infrastructure is quite important. The half adder operates by adding two single binary digits so that it can generate two outputs, a sum (S) and a carry (C). This allows the gate to perform addition, which is represented by the value of the sum that is equal to $2C + S$. Realizing this gate in its simplest form may be achieved in the parallel operation of an Exclusive-OR gate (XOR) and an AND gate (AND), with the inputs A and B directed to both gates.

The biochemical realization of this gate used enzymes to catalyze specific reactions depending on the logic of the input signals. When the different combinations were applied, they were able to selectively react with the enzymes immobilized in the system, and non-input chemicals that also flowed through the system, before being measured optically. With the separation in time and space that is inherent to a modular fluidic system, the parallel operation of the XOR and AND gates was possible.

The half-subtractor was also realized in a very similar manner, with the key difference coming from the different logic operation of the system. Unlike the binary half adder, the half subtractor operates by subtracting one single-bit binary digits from another, yielding the outputs difference (D) and borrow (Bo). This also creates a different realization. Instead of just an XOR and AND gate, the half subtractor applies an inversion (NOT) function to the AND gate's Input A , yielding an XOR and a $NOT-AND$ gate that operate in parallel.

Again, the biochemical realization of the half-subtractor is quite similar to that of the half adder, except it uses different enzymes to catalyze different reactions. This allows the chemical inputs as well as the background solution to react in a specific manner that mimics the logic of the half-subtractor. When combining these specific interactions with the separation that the fluidic infrastructure allows, the logically driven Bo and D digits to be measured separately.

Regardless of the specific enzymatic functions that allow these two systems to function, the design and implementation of this enzyme-based system illustrates how a fluidic infrastructure can simplify

the realization of a complicated computational system. With this simplification, there is the ability to create a punch-card approach, where the design of the system can be changed to suit the computational needs of the operation. Additionally, these biomolecular systems possess the capability of integration into larger infrastructures. A possible integration would be the use of the biomolecular-computing component to trigger various actuation reactions, such as a substance release process. With the possibility of these integrations, the importance of these biological computer parts is no longer relegated to the creation of a “toolbox”, but may in fact possess qualities that allow its stand-alone operation to be equally important.

Brian E. Fratto and Evgeny Katz

*Department of Chemistry and Biomolecular Science
Clarkson University, Potsdam, New York, USA*

Publication

[An Enzyme-Based Half-Adder and Half-Subtractor with a Modular Design.](#)

Fratto BE, Lewer JM, Katz E
Chemphyschem. 2016 Jul 18