Three-input majority gate using spatially localised DNA hairpins

Aby K. George ✉, *Harpreet Singh*

Department of Electrical and Computer Engineering, Wayne State University College of Engineering, Detroit, MI 48202, USA
✉ *E-mail: aby.george@wayne.edu*

Published in Micro & Nano Letters; Received on 12th September 2016; Revised on 31st October 2016; Accepted on 4th November 2016

DNA circuits employing DNA strand displacement technique are considered as a possible replacement for silicon-based circuits, especially in implantable medical devices, because of its compatibility with bio-signals, smaller size, and light weight. In this work, a new DNA strand displacement based, three-input spatially localised majority gate using DNA hairpins is proposed. The DNA hairpins used in this majority gate are tethered to an Origami substrate, and hence, the DNA hairpins are spatially separated from each other. The localisation helps the designer to reuse the strands, without causing any spurious reactions. It is believed currently that no design is available in the literature for a localised DNA majority gate. The visual DSD software is used for simulating the proposed majority gate.

1. Introduction: Molecular circuits are considered as a popular candidate for replacing silicon-based technologies for designing implantable medical circuits. With the recent advancements in DNA nanotechnology, medical devices and circuits could be built inside living organisms in a near future. Different molecular Boolean logical circuits are available in the literature, but their input and output types are different [1–4]. The output of one gate cannot be used as the input of another logic gate, and hence, such circuits are not suitable for designing a large, complex design. A DNA circuit using strands, as their inputs and output can overcome this obstacle. The power of DNA, as a computational device was first explored by Adleman, for solving the seven city Hamiltonian path problem [5]. Ever since this famous experiment, many researchers were working on DNA, to solve different problems [6–8].

Lulu Qian and Erik Winfree introduced a DNA strand displacement reaction based circuit called seesaw circuit in [9]. In [10], the scaling up of the circuit, using dual-rail AND–OR logic is discussed. Based on this dual-rail AND–OR logic, an arithmetic cell and control cell were designed [11]. The speed of operation of such circuits is limited because the entire reaction is taking place by diffusing all the molecules together. Unique strands are required for signal, and gate complexes, to avoid spurious reactions in a seesaw circuit. To solve these issues, a localised DNA strands based design approach was introduced [12, 13].

In a localised DNA circuit, the interactions of strands are limited to their neighbours. This helps the designer to reuse the strands without causing spurious reactions. The localised circuits are developed by attaching the DNA strands to a fully addressable lattice or a DNA Origami substrate [12]. The localised architecture speeds up computation in a DNA circuit, by removing much of the speed bottleneck due to diffusion. Even though different designs are available in the literature for a three-input majority gate [14, 15], none of them are using a localised architecture. In this work, we are introducing a spatially localised architecture of DNA strands for implementing a three-input majority logic operation.

2. Spatially localised DNA majority gate: A majority gate gives a logical high output when the majority of the inputs are at logical high level. For a three-input majority gate, the truth table is shown in Table 1. A, B and C are the inputs and Z is the output. The Boolean expression for the majority gate is derived from the truth table

$$Z = \mathrm{Maj}(A, B, C) = AB + AC + BC \tag{1}$$

It can be observed from the Boolean expression that the majority gate will act as a Boolean AND gate if one of the input is set to zero

$$Z = \mathrm{Maj}(A, B, 0) = AB \tag{2}$$

Similarly, by keeping one input at logical high, the majority gate will work as a Boolean OR gate

$$Z = \mathrm{Maj}(A, B, 1) = A + B \tag{3}$$

In a localised architecture, the majority logic is achieved by arranging the anchored hairpins (input translator, threshold, and output translator) in a way such that the propensity of reaction between an input translator and a threshold hairpin is greater than the propensity of reaction between an input translator and the output translator. The propensity of reaction between any two tethered strands is directly related to the distance between those strands. The propensity of reaction will be more when the strands are anchored in closer proximity to each other. A possible architecture of such a majority gate is shown in Fig. 1.

If there is no input (all inputs are low), there is no reaction and hence, no output is produced (output is low). The majority operation with only one input is present (one input is high) is shown in Fig. 1. The ^ symbol is used to identify a toehold domain and * to identify the complement of a particular domain. The input $a^\wedge s$ binds to the input translator $H(A, Y)$ with an exposed $a^{\wedge *}$, and displaces the domain $y^\wedge s$. The domain $y^\wedge s$ will bind to the free floating fuel strand $F(Y, X)$, to displace a domain $x^\wedge s$. The domain $x^\wedge s$ can bind to either the threshold $H(X, \mathit{blank})$ or the output translator $H(X, Z)$ since both of them are having an exposed $x^{\wedge *}$. However, the strands are arranged in such a manner, that the proneness of reaction between the input translator and the threshold is greater than the proneness of reaction between input translator and output translator. Hence, the domain $x^\wedge s$ binds to $H(X, \mathit{blank})$, to displace $\mathit{blank}^\wedge s$, and thus, no output is produced (output is low).

Now, consider the majority operation with two inputs present (two inputs are high) as shown in Fig. 2. The inputs A and B open the corresponding input translator hairpins to displace $y^\wedge s$. These domains bind to the fuel strands $F(Y, X)$, to open $x^\wedge s$ in both input translators. As the propensity of reaction between input translator and threshold is greater, the first $x^\wedge s$ domain binds to the threshold, to displace $\mathit{blank}^\wedge s$. The second input translator, with an open $x^\wedge s$ domain, can no longer react with the threshold. Hence, they will bind to the output translator, and displaces $z^\wedge s$, which is the output domain (output is high). A similar operation can be observed, when all the three inputs are present (all the inputs are high). The presence of first two inputs is enough to open the output domain $z^\wedge s$.

Table 1 Truth table of a three-input majority gate

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

3. Simulation results and discussion: The proposed majority gate using DNA hairpins is written in a programming language [16], implemented in the visual DSD software [17], and simulated its operation for different input combinations. An abstract diagram of the proposed three-input majority gate is shown in Fig. 3. The solid lines represent the possible reaction between the input translator strands and the threshold strand; whereas the dotted lines represent the possible reaction between the input translator strands and the output translator strand. The letters given above these lines are the tag names, indicating the relative spatial arrangements of the strands.

Each strand is associated with a set of tag names. A reaction is possible only between two neighbouring strands, which shares a common tag within an Origami substrate. It could be noted that each input translator strand shares a common tag with the threshold strand and the output translator strand. For the simulation, the local concentration of the tags, connecting the input translator strands and the threshold strands ($lc(a) = lc(b) = lc(c)$) are set to 10×10^5 nM. Similarly, the local concentration of the tags connecting the input translator strands and the output translator strand ($lc(d) = lc(e) = lc(f)$) are set to 1×10^5 nM. A high local concentration indicates that the strands sharing those tags are physically more close to each other. The propensities of reactions can be calculated by the formula given in [18]:

$$p \triangleq k \times \max(lc(a_1), lc(a_2), \ldots, lc(a_n)) \tag{4}$$

where k is the rate constant (the default DSD toehold bind rate constant is assumed to be $3 \times 10^{-4}\,\text{nM}^{-1}\text{s}^{-1}$), and $a_1, a_2, \ldots, a_n$ are the common location tags, associated with two reacting strands, for a particular reaction, in an Origami substrate.

For a three-input majority gate, the propensity of reaction between each input translator strand and the threshold strand is 300, and that of the input translator strand and the output translator strand is 30. Therefore, the input translator strand is physically closer to the threshold. Hence, the probability of a reaction between the input translator strand and threshold strand is greater, compared with the probability of a reaction between the input translator strand and the output translator strand. The visual DSD simulation traces obtained for the three-input majority gate with different input combinations are shown in Fig. 4. In this graph, different output waveforms are merged together and hence it is difficult to differentiate between them. For more clarity, the final output concentration is given for different input combinations in Fig. 5. It is evident that the output is high only when two or more inputs are high. Logical AND and OR gates are also implemented from the three-input majority gate by selecting one input as either zero or one. This is also simulated in visual DSD.

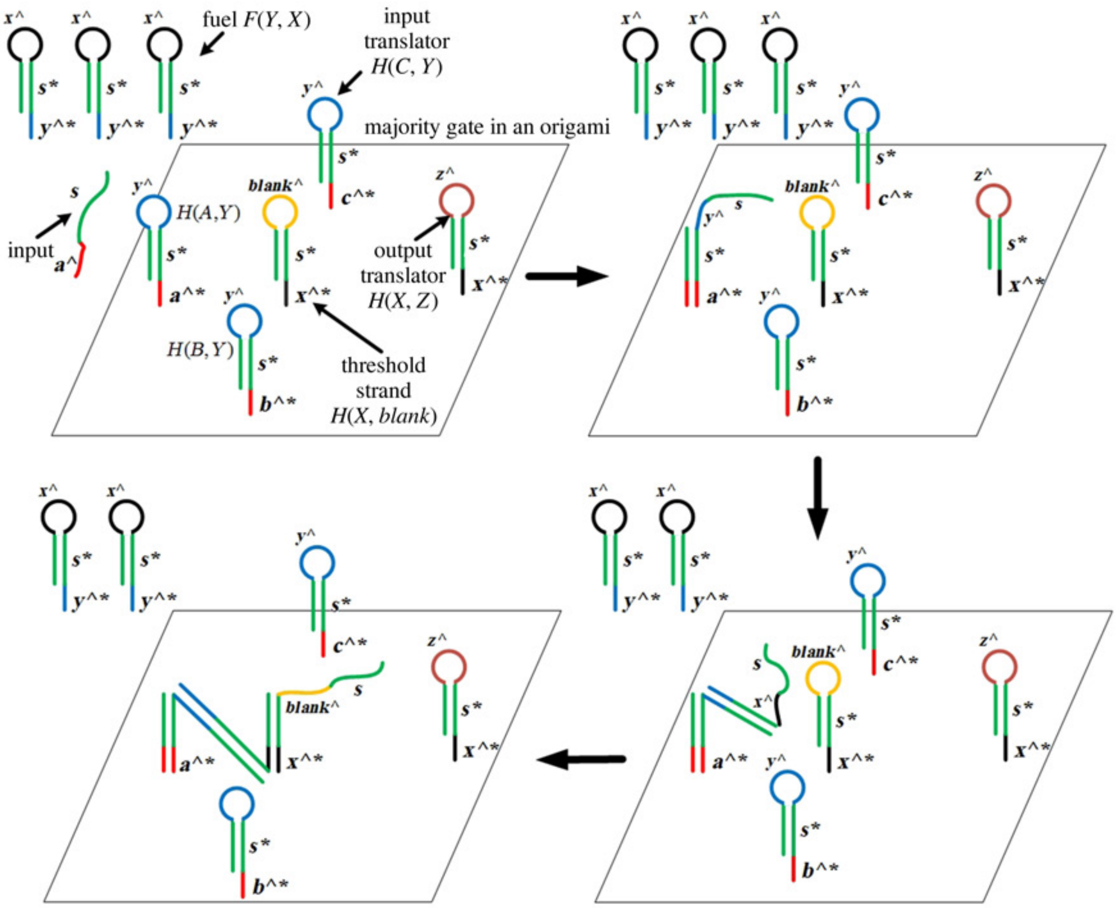

Fig. 1 *DNA hairpin strand operation for localised majority gate with only one input present*

Micro & Nano Letters, 2017, Vol. 12, Iss. 3, pp. 143–146
doi: 10.1049/mnl.2016.0535

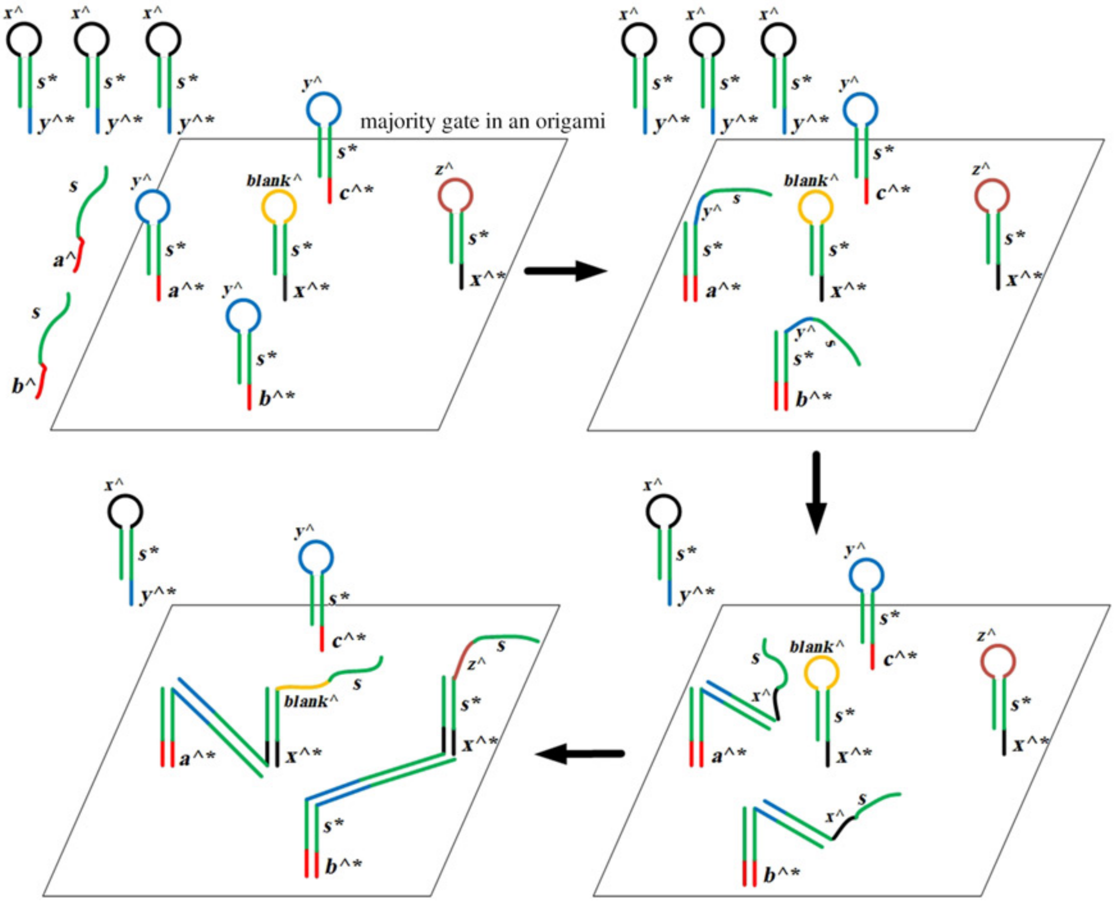

Fig. 2 *DNA hairpin strand operation for localised majority gate with two inputs present*

The design of existing DNA three-input majority gates can be found in [14, 15]. The first design [14] uses different types of input strands and have three AND and one OR operations. The input and outputs are not uniform and hence the scaling up of the circuit is difficult. Also, the number of strands used in the design is more compared with the proposed spatially localised design. In the second design [15], a central circular DNA strand structure is used to make a three-input majority gate operation. This design is prone to the scaling up issues such as the requirement of a large number of unique strands to avoid spurious reactions, and limited speed of operation since all the molecules are diffused together. These issues could be solved by using the new localised majority gate proposed in this Letter.

The DNA strand displacement based circuits cannot produce a NOT operation, hence a dual-rail logic is required. The designs available in the literature for scaling up of circuits concentrate on dual-rail AND–OR circuits [10]. In a dual-rail circuit, one signal is represented by the signal itself and its complement. For example, the signal A can be represented in dual-rail logic as $A_1 = A$ and $A_0 = \overline{A}$. The majority gate is a good candidate for designing dual-rail circuits. The majority gate is having the following property

$$\overline{M(A, B, C)} = M(\overline{A}, \overline{B}, \overline{C}) \tag{5}$$

In dual-rail logic, (A, B, C) can be represented by (A_1, B_1, C_1), and

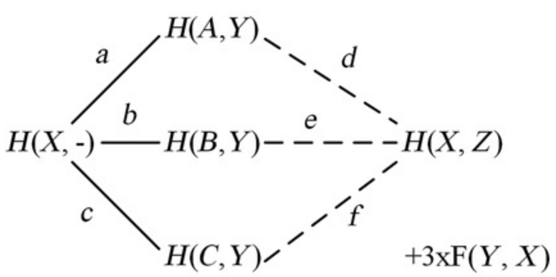

Fig. 3 *Abstract diagram of a three-input majority gate with labels showing interaction between hairpins*

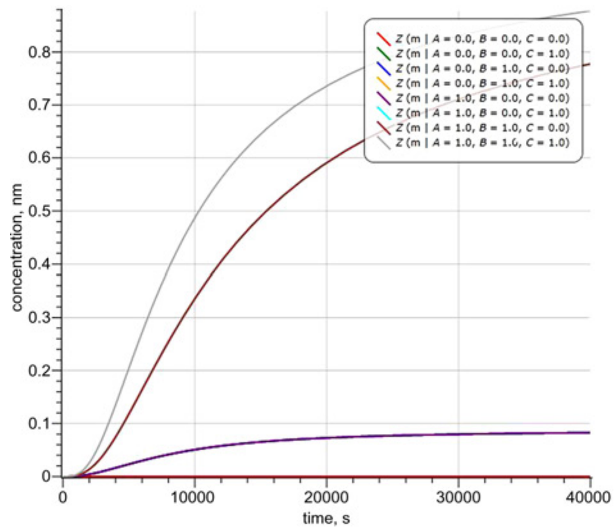

Fig. 4 *Time courses for different input combinations in visual DSD*

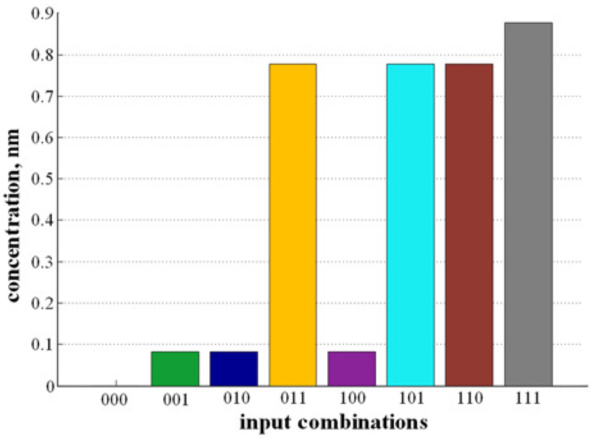

Fig. 5 *Final concentrations obtained for a three-input majority gate for different input combinations*

$(\bar{A}, \bar{B}, \bar{C})$ by (A_0, B_0, C_0) then

$$\begin{aligned} M(A, B, C) &= M(A_1, B_1, C_1) \\ \overline{M(A, B, C)} &= M(A_0, B_0, C_0) \end{aligned} \tag{6}$$

With the development of localised majority gate, it is possible in the future to design either dual-rail majority logic or a dual-rail AND–OR–majority logic for complex circuits. Thus the availability of majority gate will introduce more flexibility while designing complex circuits.

The design presented in this Letter is a simulation model. The experimental validation is still a challenge. However Muscat *et al.* [13] were able to implement the translation operation with a three-step transmission line. There are many system parameters such as sequence design, physical positioning of the strands, the length of the domains and so on, which will play an important role while experimentally implementing the proposed circuit.

4. Conclusion: In this work, we are theoretically implementing a novel three-input majority gate, which uses spatially localised DNA hairpin strands attached to an Origami substrate. We synopsised the benefits of localisation of the majority gate and explained the working of DNA strand displacement operations taking place in a majority gate under different conditions. The three-input majority gate is implemented in visual DSD software, and simulated the design for different input combinations. A three-input majority gate can function as logical AND and OR gates. A possible scaling-up of circuits using dual-rail AND–OR–majority logic is also discussed. The availability of a three-input majority gate will give more flexibility for the designer while developing the synthesising tools for DNA digital design.

5. Acknowledgments: The authors are very thankful to the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

6 References

[1] Gentili P.L.: ‘The fundamental fuzzy logic operators and some complex Boolean logic circuits implemented by the chromogenism of a spirooxazine’, *Phys. Chem. Chem. Phys.*, 2011, **13**, (45), pp. 20335–20344

[2] Gentili P.L.: ‘Molecular processors: from qubits to fuzzy logic’, *ChemPhysChem*, 2011, **12**, (4), pp. 739–745

[3] Wu Q.Q., Duan X.Y., Song Q.H.: ‘Resettable multiple-mode molecular arithmetic systems based on spectral properties of 2-quinolin-2-ylmethylene-malonic acids’, *J. Phys. Chem. C*, 2011, **115**, (48), pp. 23970–23977

[4] Gentili P.L., Horvath V., Vanag V.K., ET AL.: ‘Belousov–Zhabotinsky ‘chemical neuron’ as a binary and fuzzy logic processor’, *IJUC*, 2012, **8**, (2), pp. 177–192

[5] Adleman L.M.: ‘Molecular computation of solutions to combinatorial problems’, *Science*, 1994, **266**, (5187), pp. 1021–1024

[6] Kari L., Gloor G., Yu S.: ‘Using DNA to solve the bounded post correspondence problem’, *Theor. Comput. Sci.*, 2000, **231**, (2), pp. 193–203

[7] Adleman L.M.: ‘Computing with DNA’, *Sci. Am.*, 1998, **279**, (8), pp. 34–41

[8] Ouyang Q., Kaplan P.D., Liu S., ET AL.: ‘DNA solution of the maximal clique problem’, *Science*, 1997, **278**, (5337), pp. 446–449

[9] Qian L., Winfree E.: ‘A simple DNA gate motif for synthesizing large-scale circuits’, *J. R. Soc. Interface*, 2011, **8**, pp. 1281–1297

[10] Qian L., Winfree E.: ‘Scaling up digital circuit computation with DNA strand displacement cascades’, *Science*, 2011, **332**, (6034), pp. 1196–1201

[11] George A.K., Almatrood A., Singh H.: ‘Design of arithmetic and control cells for a DNA binary processor’. Proc. Int. Conf. on Computational Science and Computational Intelligence (CSCI), 2015, pp. 7–12

[12] Chandran H., Gopalkrishnan N., Phillips A., ET AL.: ‘Localized hybridization circuits’. Proc. Int. Workshop on DNA-Based Computers, 2011, pp. 64–83

[13] Muscat R.A., Strauss K., Ceze L., ET AL.: ‘DNA-based molecular architecture with spatially localized components’, *ACM SIGARCH Comput. Archit. News*, 2013, **41**, (3), pp. 177–188

[14] Zhu J., Zhang L., Dong S., ET AL.: ‘Four-way junction-driven DNA strand displacement and its application in building majority logic circuit’, *ACS Nano*, 2013, **7**, (11), pp. 10211–10217

[15] Li W., Yang Y., Yan H., ET AL.: ‘Three-input majority logic gate and multiple input logic circuit based on DNA strand displacement’, *Nano Lett.*, 2013, **13**, (6), pp. 2980–2988

[16] Phillips A., Cardelli L.: ‘A programming language for composable DNA circuits’, *J. R. Soc. Interface*, 2009, **6**, (Suppl 4), pp. S419–S436

[17] Lakin M.R., Youssef S., Polo F., ET AL.: ‘Visual DSD: a design and analysis tool for DNA strand displacement systems’, *Bioinformatics*, 2011, **27**, (22), pp. 3211–3213

[18] Lakin M.R., Petersen R., Gray K.E., ET AL.: ‘Abstract modelling of tethered DNA circuits’. Proc. Int. Workshop on DNA-Based Computers, 2014, pp. 132–147

doi: 10.1049/mnl.2016.0535