

# Circuit design for static genetic memory

Chun-Liang Lin, Yu-Jia Hu, Wei-Xian Li

Department of Electrical Engineering, National Chung Hsing University, Taichung, 402, Taiwan  
E-mail: chunlin@dragon.nchu.edu.tw

Published in *The Journal of Engineering*; Received on 8th August 2017; Accepted on 24th September 2017

**Abstract:** As in traditional computers, future bio-computers might feature the analogous mechanism to carry out a set of arithmetic or logical operations sequentially. In digital computers, memory is essential for which digitised information is encoded, stored, and retrieved. A stored memory unit in the computer is almost constructed by metal–oxide–semiconductor field-effect transistors. In the future, realising the specific data stored memory unit in the biocomputer is essential. However, it is not easy to realise that mechanism for data memorisation in biological systems. The authors propose a structure of a fundamental memory unit necessary with input/output configuration for data read and write for the biocomputer.

## 1 Introduction

Synthetic biology is typically defined as the combination of engineering and biology in order to design and realise biological functions. The aim is to streamline the practice of biological engineering, to shorten the time required to design, build, and test synthetic gene networks. The discipline mostly concentrates on understanding the behaviours of the biological system, trying to create a specific artificial genetic circuit based on the principles of systems biology, mathematics, and engineering [1–3]. The synthetic genetic circuit includes individual biological parts to realise the biochemical process of organisms and to achieve a specific function [4–6]. By integrating a variety of biological components, sophisticated bio-computing modules can be expected, like the very-large-scale integration of silicon components into current computers.

The customised genetic circuits with specific functions can be synthesised by considering mathematical models to capture the real behaviour from the quantitative or qualitative viewpoints. Inspired by electronic circuits, several synthetic genetic circuits have recently been created [7], such as repressilator [8], toggle switch [9], genetic oscillator [10, 11], genetic feedback controllers [12], logic evaluator, and cell–cell communication [13–15]. A toggle switch capable of controlling metabolism by stepwise function was successfully created, which is constructed from two irrepressible promoters arranged in a mutually inhibitory network [13]. In [4, 8], the authors developed oscillatory circuits using multiple self-regulating mechanisms to create a time-dependent oscillation of gene product expression. A limited counting mechanism was recently implemented with pulse-controlled gene circuits, which enables genetic ‘programming’ of cells in the work [16].

A Boolean logic gate is a fundamental and essential unit in the digital logic circuits. To associate the digital logic design with biological systems, a direct way is to combine a variety of genetic logic gates to realise the complex biological logic counterparts. Different genetic transcriptional reactions have been utilised to express logic gates with specific functions in [17–22]. Many biological logic behaviours have been well developed in the existing studies, such as NOT, Buffer, AND, OR, XOR, and NOR. Biochemists constructed chemical reactions from the logic phenomenon that responses cellular and environmental molecular signals in vivo. These chemical reactions can be represented in a set of differential equations. Through the cascaded connection of individual genetic logic gates, a class of logic circuits with advanced functions can be synthesised accordingly, such as a multiplexer, a half adder, combinational logic circuits, memory, and sequential logic circuits

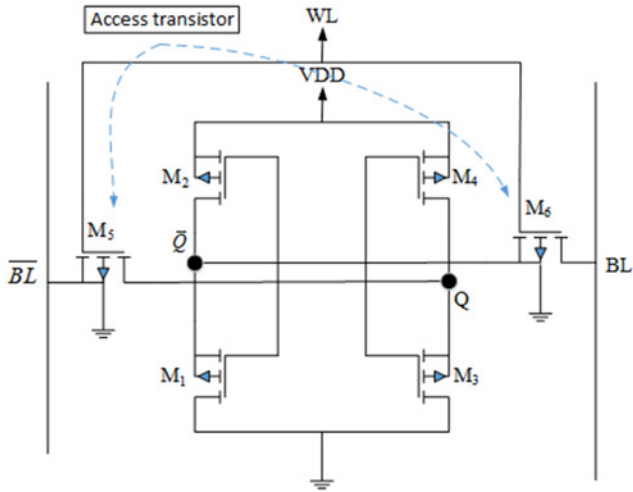
[23–27]. Recently, combining a clock signal with a genetic sequential logic circuit, advanced biologic devices such as the SR latch and flip-flop (FF) can be realised [25–28]. In general, engineering biology requires integrated design across many subcellular systems. For genetic circuit design automation, Prof. Voigt and his team members have developed the software tool ‘Cello’ to automate the selection and concatenation of biological parts and balance the associated constraints [29]. It enables rapid design of large-scale multipart systems.

In digital electronics, combinational logic circuits are very fundamental, which are digital circuits used to realise certain Boolean functions or many others. Unlike sequential logic circuits, whose outputs depend on both the input and their previous output state, the outputs of combinational logic circuits are determined by the logic function of their current input level, logic 0 or 1, at any given instant of time. Combinational logic, also known as time-independent logic, can be used to perform a specific Boolean algebra. As for the kernel of current computers, the arithmetic logic unit is constituted by a lot of combinational logic circuits.

Nowadays, there is a growing demand for digital storage capacity. Static random access memory (SRAM) is one of the data storage units in the modern computers. In Fig. 1, it uses a bistable latch circuit to store data bits. One bit SRAM is typically constituted by six metal–oxide–semiconductor field-effect transistors (MOSFETs), two of which are used as access switches. The MOSFET has an insulated gate, whose voltage determines the conductivity of the device. Changing conductivity with the amount of applied voltage is used for amplifying or switching current passed through. The four MOSFETs, M1–M4, form the core to store a data bit in the form of the voltage level. The word line (WL) is connected to the gate of the two transistors, and two access transistors, M5 and M6, are used to control access to the memory core during a read/write (R/W) operation. Two access transistors are used as the switch. When the WL is at a high (low) level, M5 and M6 will be on (off), BL and  $\overline{\text{BL}}$  are used as the data transmission channels for the R/W operation during the period.

When a high (low) voltage level is added at BL then a data 1 (0) is stored to  $Q$ . The coupled connection of M1–M4 hold the status of  $Q$  and  $\overline{Q}$  forming the function of data memory.

The idea of using DNA as a storage device was briefly mentioned in [30, 31]. However, to the best knowledge of the authors, the study of biological memory is very rare. The paper here is an extension to our preliminary study [32]. We have proposed a possible genetic data storage unit for future biotech computers. The basic genetic logic gate is combined with the genetic sequential logic circuit to realise a biological SRAM. Each bit cell consists of two



**Fig. 1** Fundamental structure of one bit SRAM in electronic computer (M1–M6 are MOSFETs)

biological D latches as a switch that allows binary data to access the memory core. The two head-to-tail connected inverters hold the bits of each memory core. An extensive simulation study has been conducted to verify the feasibility of the proposed design.

## 2 Method

### 2.1 Model of synthetic genetic logic circuits

A synthetic genetic circuit can be realised in a specific function by using mathematical models that describe the biochemical reactions in the biological system. The following dynamic equations describe the dynamic model of the genetic logic circuit with  $L$  genes [15]:

$$\begin{aligned}\dot{m}_i &= \alpha_i f_i(u) - \lambda_i m_i + \alpha_{i,0} + I_i, \\ \dot{p}_i &= \beta_i m_i - \gamma_i p_i, \quad i = 1, \dots, L\end{aligned}\quad (1)$$

where  $m_i \in R_+$  denotes the concentration of mRNA,  $p_i \in R_+$  is the  $i$ th protein,  $\alpha_i \in R$  is the transcription rate of the mRNA,  $\lambda_i$  is the degradation rate of the mRNA,  $\gamma_i$  is the degradation of the protein,  $\beta_i$  is the synthesis rate of the protein,  $\alpha_{i,0}$  denotes the basal production rate. The promoter activity function  $f_i(\cdot)$  specifies the non-linear transcriptional reactions between the regulated protein and the RNA polymerase (RNAP).  $I_i$  denotes the input,  $u$  denotes the concentration of the transcription factor (TF) from other gene's products, or inducers that control the gene expression.

For the model of a gene with a single activator TF, such as a NOT, the promoter activity functions are given by [24, 27]

$$f_{\text{NOT}}(u) = \frac{1}{1 + (u/K)^n}, \quad (2)$$

where  $f_{\text{NOT}}$  is the promoter activity function for the logic NOT [24, 27],  $n \in R$  is the Hill coefficient denoting the binding cooperativity between the TF and the corresponding operator, and  $K$  is the Hill constant. For the logic NOT gate, the input is a repressor and the gene produces protein only in the absence of the repressor; otherwise, the presence of the repressor would obstruct the bound of RNAP and the promoter.

For the genes with two operator sites, which can bind two repressor TFs or activator TFs, their promoter activity functions can be

described in accordance with the corresponding logic functions as

$$f_{\text{AND}}(u_1, u_2) = \frac{(u_1/K_1)^{n_1} (u_2/K_2)^{n_2}}{1 + (u_1/K_1)^{n_1} + (u_2/K_2)^{n_2} + (u_1/K_1)^{n_1} (u_2/K_2)^{n_2}} \quad (3)$$

and

$$f_{\text{NOR}}(u_1, u_2) = \frac{1}{1 + (u_1/K_1)^{n_1} + (u_2/K_2)^{n_2} + (u_1/K_1)^{n_1} (u_2/K_2)^{n_2}}, \quad (4)$$

where  $f_{\text{AND}}$  and  $f_{\text{NOR}}$  are, respectively, the promoter activity functions of AND and NOR gates;  $u_1$  and  $u_2$  are concentrations of the repressor or the activator of the TFs;  $K_1$  and  $K_2$  are the Hill constants for  $u_1$  and  $u_2$ , respectively;  $n_1$  and  $n_2$  are the respective corresponding to the Hill coefficients. For the logic AND (NOR) gate, its transcriptional behaviours are regulated by two activator (repressor) TFs with different binding sites.

### 2.2 Dynamic model of synthetic genetic logic circuits

Two head-to-tail connected inverters form the kernel (per bit) of the SRAM. If the same  $K_i$ ,  $\alpha_i$ ,  $\beta_i$  and  $\lambda_i$  for the two inverters are used, the basal production rate is comparatively ignorable, and  $I$  is a normalised impulse signal denoting the data '1' or zero for the data '0', its dynamic equation can be expressed as

$$\begin{aligned}\dot{m}_1 &= \alpha f_{\text{NOT}}(p_2) - \lambda m_1 + I, \\ \dot{p}_1 &= \beta m_1 - \gamma p_1, \\ \dot{m}_2 &= \alpha f_{\text{NOT}}(p_1) - \lambda m_2, \\ \dot{p}_2 &= \beta m_2 - \gamma p_2\end{aligned}\quad (5)$$

Since the response of mRNA is far faster than that of the protein, we consider the steady-state behaviour of the mRNA and obtain

$$\begin{aligned}\dot{p}_1 &= \frac{\beta}{\lambda} [\alpha f_{\text{NOT}}(p_2) + I] - \gamma p_1, \\ \dot{p}_2 &= \frac{\beta}{\lambda} [\alpha f_{\text{NOT}}(p_1)] - \gamma p_2.\end{aligned}\quad (6)$$

If  $(p_i/K) \ll 1$ , it can be approximated by

$$\begin{aligned}\dot{p}_1 &= -\gamma p_1 - \frac{\alpha\beta}{\lambda} \left( \frac{p_2}{K} \right)^n + \frac{\alpha\beta}{\lambda} \left( 1 + \frac{I}{\alpha} \right), \\ \dot{p}_2 &= -\gamma p_2 - \frac{\alpha\beta}{\lambda} \left( \frac{p_1}{K} \right)^n + \frac{\alpha\beta}{\lambda}.\end{aligned}\quad (7)$$

Considering steady state behaviours, we set

$$\frac{\alpha\beta}{\lambda\gamma} = 1. \quad (8)$$

When  $p_2 = 0$ , it gives  $p_1 = 1$ . On the other hand, when  $p_1 = 0$  then  $p_2 = 1$ . The outputs of the two NOT gates are complementary as desired

### 2.3 Description of D-latch

The SR-latch is a circuit with two cross-coupled NOR gates and consists of two inputs, respectively, the label S is for the set and R for the reset, and two corresponding outputs  $Q$  and its inverse  $\bar{Q}$ . The construction framework and its corresponding truth table are shown in Fig. 2.

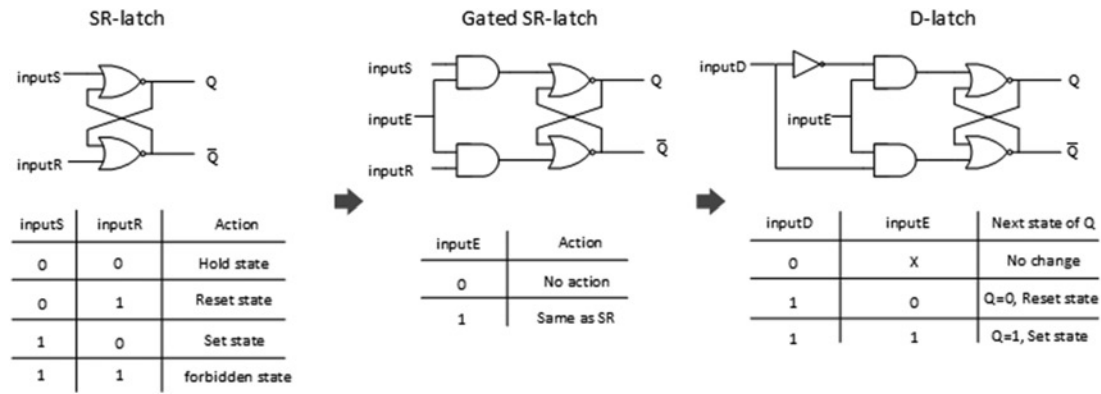


Fig. 2 Implementation of D-latch from SR latch

The D latch is a FF that has only one protein input (D), one for enabling the control input (E) of the FF and two outputs  $Q$  and  $\bar{Q}$ . The D latch eliminates the undesirable condition of the SR latch. The truth table depicts that when the input E is logic low (L), the outputs  $Q$  and  $\bar{Q}$  are latched in their previous states regardless of the state of the data input D. When E is logic high (H), the switch is enabled, the output will follow the data input (D is logic H (L), then  $Q$  is logic H (L)). In [27], it has developed a biological D-latch and other representative biological gates. The equivalent circuit diagram from the biological viewpoint can be illustrated as in Fig. 3.

The dynamic behaviour of the genetic D-latch is modelled by the following dynamic equations:

$$\begin{aligned}
 \dot{p}_{\text{NOT}} &= \alpha_p f_{\text{NOT}}(D) - \gamma_p p_{\text{NOT}} + \alpha_{p0, \text{NOT}}, \\
 \dot{p}_{\text{AND1}} &= \alpha_p f_{\text{AND1}}(p_{\text{NOT}}, E) - \gamma_p p_{\text{AND1}} + \alpha_{p0, \text{AND1}}, \\
 \dot{p}_{\text{AND2}} &= \alpha_p f_{\text{AND2}}(D, E) - \gamma_p p_{\text{AND2}} + \alpha_{p0, \text{AND2}}, \\
 \dot{p}_{\text{NOR1}} &= \alpha_p f_{\text{NOR1}}(p_{\text{AND1}}, \bar{Q}) - \gamma_p p_{\text{NOR1}} + \alpha_{p0, \text{NOR1}}, \\
 \dot{p}_{\text{NOR2}} &= \alpha_p f_{\text{NOR2}}(p_{\text{AND2}}, Q) - \gamma_p p_{\text{NOR2}} + \alpha_{p0, \text{NOR2}},
 \end{aligned} \quad (9)$$

where  $p_{\text{NOT}}$ ,  $p_{\text{AND1}}$ ,  $p_{\text{AND2}}$ ,  $p_{\text{NOR1}}$ , and  $p_{\text{NOR2}}$  are, respectively, the output concentrations of the NOT gate, the first AND gate, the second AND gate, the first NOR gate, and the second NOR gate.

The biological D-latch becomes active only when the input D goes from low to high. The NOT gate produces protein only in the absence of the input D. The proteins  $p_{\text{NOT}}$  and  $p_E$  activate the transcription of the gate AND<sub>1</sub>. The proteins  $p_D$  and  $p_E$  activate the transcription of the gate AND<sub>2</sub>.  $p_{\text{AND1}}$  and  $p_{\bar{Q}}$  inhibit transcription of the gate NOR<sub>1</sub> and  $p_{\text{AND2}}$  and  $p_Q$  inhibit transcription of the gate NOR<sub>2</sub>. The associated parameters are defined as in (1).

#### 2.4 Design of biological SRAM

Inspired by the fundamental structure of one bit silicon SRAM depicted in Fig. 1, we propose a functionally equivalent genetic SRAM. The equivalence between electronic and biological operations can be found in Fig. 4. There is no counterpart developed in the field of system biology to a transistor in the semiconductor industry so far. We propose to use two D-latches (D<sub>1</sub>-latch and D<sub>2</sub>-latch) to replace the two access transistors to control data transfer. The sequential logic operation of the D latch is used as a genetic switch. They provide two head-to-end connected inverters and bit line (BL) joins. The head-to-end connection of the  $I_{\text{nv}_1}$  and  $I_{\text{nv}_2}$  are connected to the inverter to keep the inverter output due to the positive feedback.

In the electronic circuits, D-latch can be used to fulfil the function of storage, when the enable E is at the high level, the new data is written into the memory element; when E is at the low level, the output  $Q$  remains the state when E is logic high before. This is the fundamental function of a memory unit. However, it was not

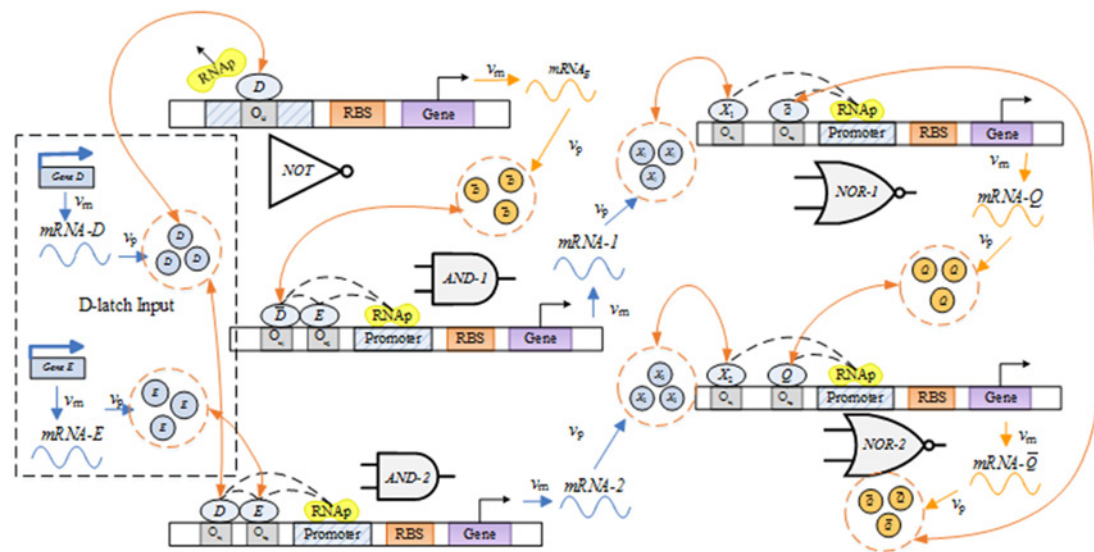


Fig. 3 Structure of the biological D-latch

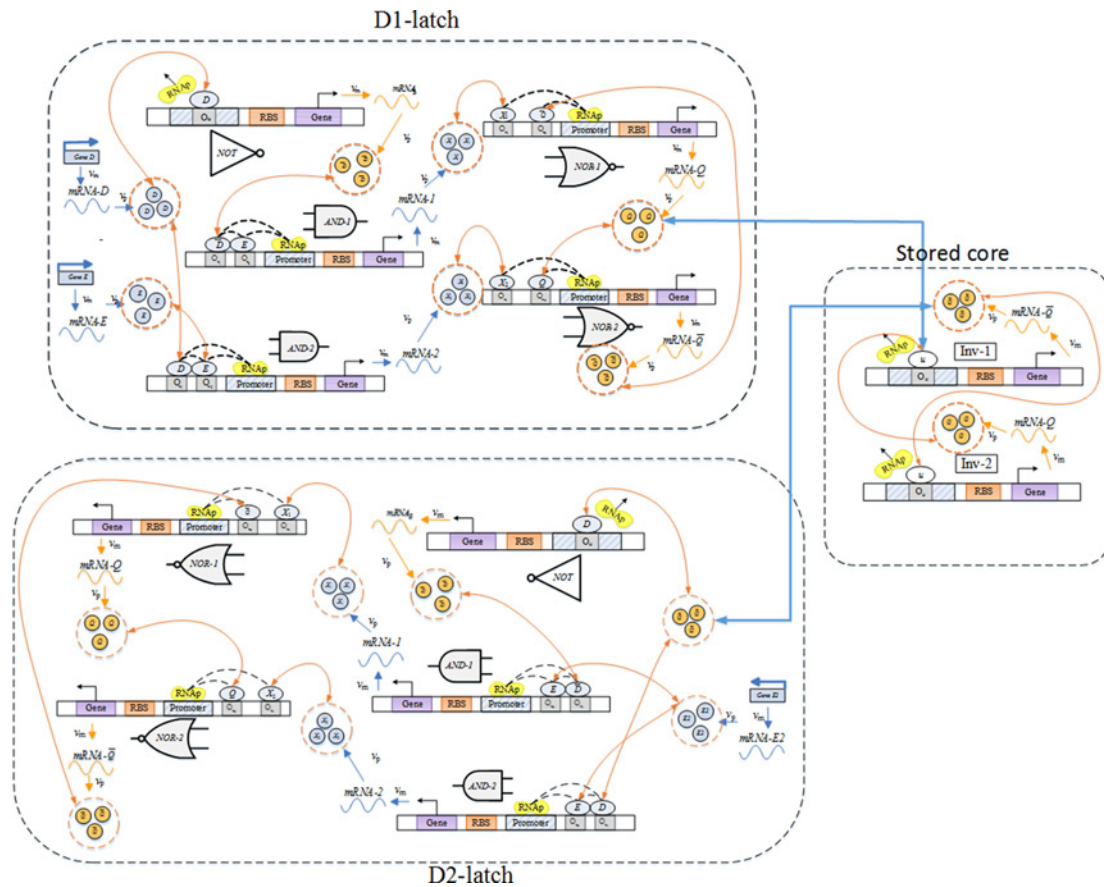


Fig. 4 Structure of one bit biological SRAM

used as a memory in practice for two reasons. Complementary metal–oxide–semiconductor implementation of the D latch needs more transistors than the SRAM cell, and the D-latch has the problem of current leakage, the leakage will eventually destroy the state if the state remains for a long time. Without using the D-latch as the memory core, we adopt here a bistable structure formed by two head-to-tail connected inverters, the state will not stay in the astable state indefinitely, once they enter the astable state, the state will immediately regain one of the two stable states and stay for long.

The biological SRAM unit possesses three operational modes explained below. In this model,  $WL_1$  corresponds to ‘input- $E_1$ ’,

$BL$  corresponds to ‘input- $D_1$ ’,  $WL_2$  corresponds to ‘input- $E_2$ ’ and  $Q_1$  corresponds to ‘input- $D_2$ ’.

**Standby:** in this operation mode, the  $WL$  is not set that turns off two access transistors. The data cannot be stored in the memory kernel. The corresponding biochemical reaction is shown in Fig. 5.

**Data read:** the control unit needs to impose logic H to the  $WL$ . The memorised state of the SRAM is read via the access transistor and bit line. The read cycle starts by charging the high protein concentration on  $WL$ , enabling the D-latch and causing the state of the two head-to-tail connected inverters transferred. See Fig. 6.

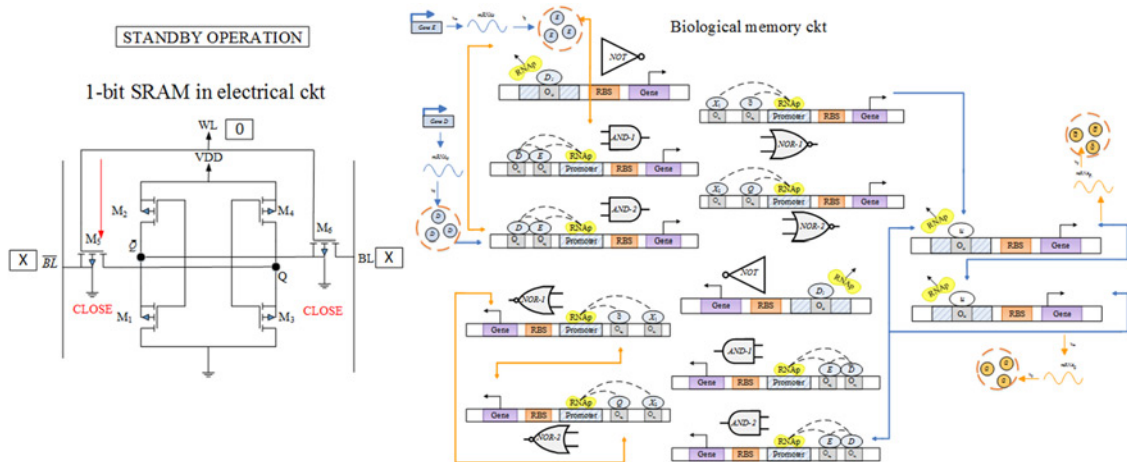


Fig. 5 Equivalence between electrical and biological circuits in STANDBY operation

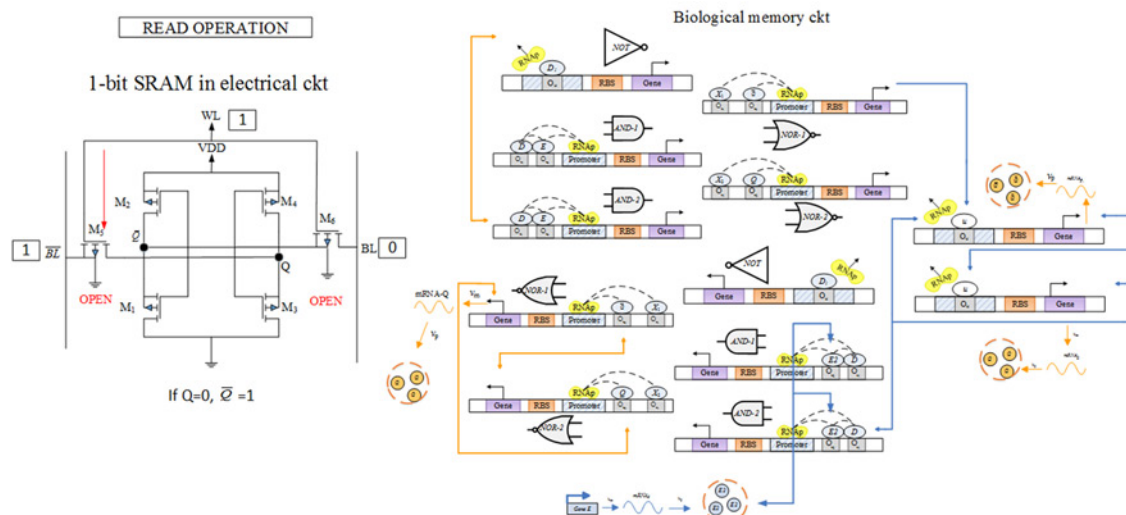


Fig. 6 Equivalence between electrical and biological circuits in READ operation

**Data write:** it starts with BL writing data. We apply logic L to BL and logic H to the input E to enable the action. This will cause the D-latch to change the state. The information from the BL to the two cross-coupled inverters will remain. Alternatively, it applies logic L to input E, changes the state from enable to invariant, and ends the operation. See Fig. 7.

For demonstration, the sequential operation of the SRAM memory cell is depicted in Figs. 8 and 9.

The operation starts from the writing process. During this phase, D<sub>1</sub>-latch opens and D<sub>2</sub>-latch closes. Fig. 8a shows the normalised input commands of D and E at the first D-latch. When the enable command is high, the data will be transferred to the first D-latch.

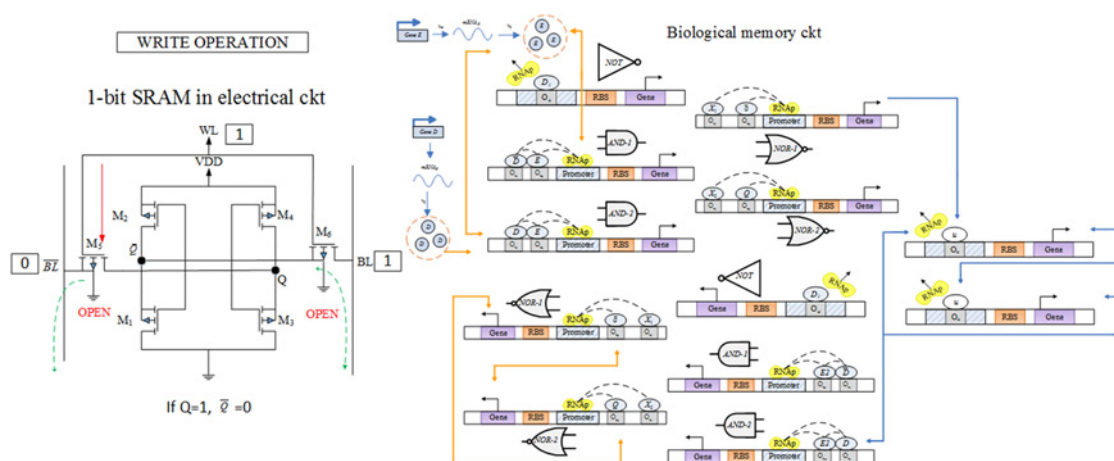


Fig. 7 Equivalence between electrical and biological circuits in WRITE operation

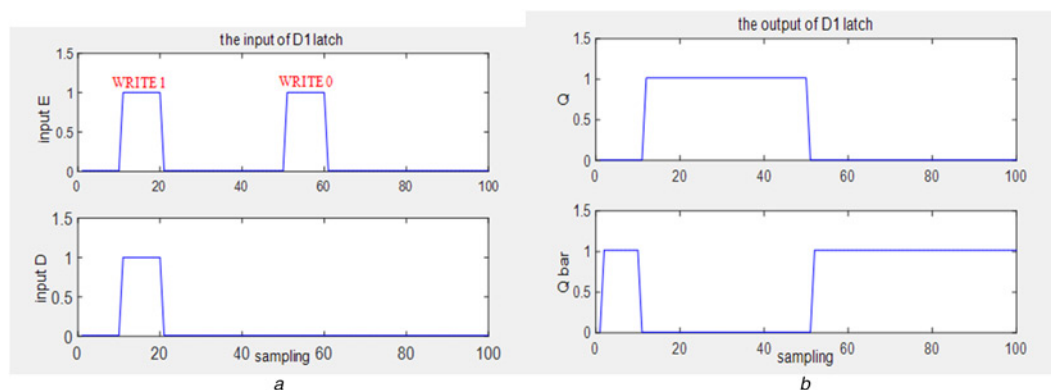
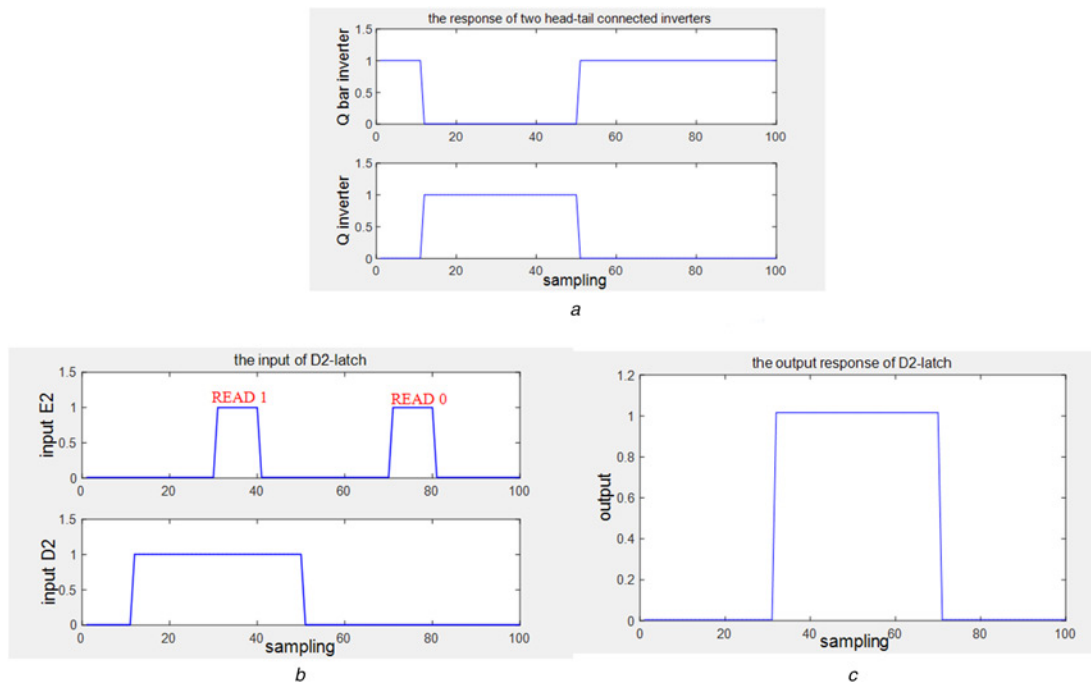


Fig. 8 Input response and output response of the first D-latch  
a Normalised input commands of D and E of the first D-latch  
b Normalised outputs  $Q$  and  $\bar{Q}$  of the first D-latch



**Fig. 9** Inverter response and the second D-latch's input response, output response  
*a* Normalised responses of  $Q$  and  $\bar{Q}$  during the standby status  
*b* Normalised input commands of D and E of the second D-latch  
*c* Normalised output response of the second D-latch

Fig. 8*b* shows the normalised output responses  $Q$  and  $\bar{Q}$  of  $D_1$ -latch.

When both  $D_1$  and  $D_2$ -latches are closed, the operation enters the standby phase. Two head-to-tail connected inverters hold data transferred from the first D-latch. Fig. 9*a* illustrates the normalised output responses  $Q$  and  $\bar{Q}$  at this phase.

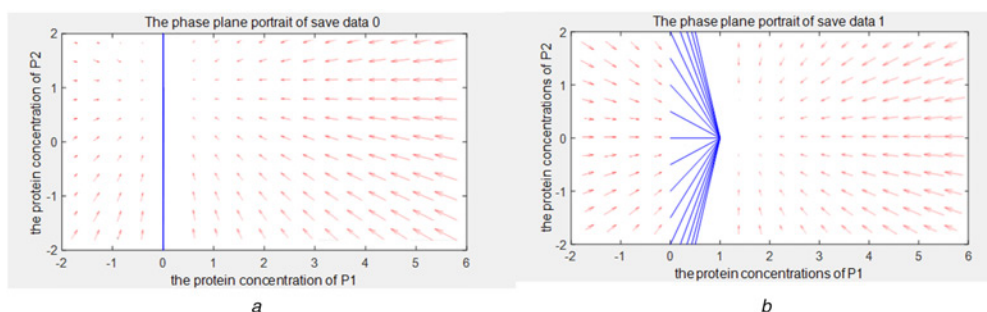
The reading phase works when  $D_1$ -latch closes and  $D_2$ -latch opens. Data stored within the two inverters is sent to the input port D of  $D_2$ -latch. Fig. 9*b* shows the simulation result with the input command at D and the enable command at E of  $D_2$ -latch. Fig. 9*c* displays data readout (i.e. the responses  $Q$  of  $D_2$ -latch) which is consistent with the content of the memory kernel stored at  $I_{nv,2}$ .

In practice, a complete memory unit should be constituted by a large amount of memory cell. When the module size increases with the need of more bits, the systems cannot just be placed in a cell without having negative effects on the well-being of the cell. The insulation of biological parts from each other was not considered because the system level design was focused. This could be resolved by distributing the circuits among a population of cells for insulation; however, then the problems with communication

and synchronous behaviour must be tackled. In practice, examination of potential cross-talk among biological parts should not be ignored. To avoid the problem, transcriptional insulation should be fulfilled, which can be achieved for gates by adding a different strong terminator with sufficiently diverse sequences to avoid homologous recombination [29]. The output promoters should also be insulated on both sides from changes to their upstream and downstream context so that a promoter generates the same response function irrespective of the downstream gene. One way to solve the issue is to add a biological buffer in between two circuits [33] which functionally serves as the role of a buffer in the electronic circuit realised by op-amp of the unit gain. That provides an electrical impedance transformation from one circuit to another, with the aim of the signal source being unaffected by whatever currents (or voltages, for a current buffer) that the load may produce.

## 2.5 Analysis of biological SRAM

In the non-linear system analysis, the phase plane portrait is a useful method to determine stability feature around the equilibrium point.



**Fig. 10** Phase plane portrait for data storage  
*a*  $(p_1, p_2) = (0, 1)$   
*b*  $(p_1, p_2) = (1, 0)$

To sketch solutions in the phase plane of the current problem we pick up a sequence of  $t$  and the corresponding state value of  $(p_1, p_2)$ . Doing this allows one to sketch state trajectory (solution behaviour) in the phase plane. We plot the solution of the differential equation given by (6) and the vector field on the coordinate plane with the values of the two-state variables. The vector field displayed in Fig. 10 illustrates how the solutions to the two equations will go from the given initial state. Fig. 10a displays the situation when data 0 is stored in the SRAM storage cell. This is equivalently realised by setting  $\alpha_{p0,NOT1} = 1$  and  $\alpha_{p0,NOT2} = 0$ . It is seen that the normalised protein concentration of  $p_1$  and  $p_2$  converge to (0, 1). As a counterpart, Fig. 10b shows the situation when data 1 is stored, the concentrations of  $p_1$  and  $p_2$  converge to (1, 0).

### 3 Discussions

SRAM and dynamic RAM (DRAM) are two types of popular memory. They are different from function and structure. The SRAM storage cell involves two head-to-tail connected inverters forming a positive feedback connection meaning that the bistable latching circuit does not need to be refreshed. This will hold the data as long as power is on. Different from the SRAM, a DRAM uses a transistor as the switch and a capacitor as the storage cell. The charging status of the capacitor is used to symbolise the data storage per bit or zero. However, due to the physical characteristics, capacitors may have a small amount of leak current when energised. Leakage current is a result of electrons physically making their way through the dielectric medium, around its edges or across its leads. This will over time fully discharge the capacitor when the supply voltage vanishes. On comparing with DRAM, a storage unit of the SRAM needs six transistors, thus, the DRAM possesses higher density but with a lower response speed. Our research proposes the SRAM structure in the biological sense, which has an advantage of a faster response and need not be dynamically refreshed. However, the memory size is comparatively large. How to reduce the size per storage cell is an interesting issue worthy of further development which is currently under study.

### 4 Conclusions

The construction of a SRAM based on biological genetic logic circuits is proposed in this research. The biological circuit is formed by a biological inverter comprising two tail-head connections as a memory core for holding the state with two D-latches serving as the access switch to control data read and write. In the electronic circuits of the traditional computers, a single bit data is stored using the state of a six-transistor memory cell involving two access transistors to control data transmission. However, as far as the authors are aware, no bio-switch like transistor has been attempted. From the electronic viewpoint, the switching operation is realised by using the current or voltage transfer of the physical characteristics of the semiconductor. In the biological world, no genetic device can produce similar functions. We have changed the symmetrical R/W structure to D-latch for controlling data transmission in the current configuration. For the memory core, two head-to-tail connected inverters constitute the feedback loop for data holding. The configuration is shown to be effective. Experimental results *in silico* show satisfactory dynamic responses of the four-bit data memory.

### 5 References

- [1] Cuccato G., Gatta G.D., di Bernardo D.: 'Systems and synthetic biology: tackling genetic networks and complex diseases', *Heredity*, 2009, **102**, pp. 527–532
- [2] Khalil A.S., Collins J.J.: 'Synthetic biology, applications come of age', *Nat. Rev. Genetics*, 2007, **11**, pp. 367–379
- [3] Ruder W.C., Lu T., Collins J.J.: 'Synthetic biology moving into the clinic', *Science*, 2011, **333**, pp. 1248–1252
- [4] Sprinzak D., Elowitz M.B.: 'Reconstruction of genetic circuits', *Nature*, 2005, **438**, pp. 443–448
- [5] Ichinose N., Yada T., Gotoh O., *ET AL.*: 'Reconstruction of transcription-translation dynamics with a model of gene networks', *J. Theor. Biol.*, 2008, **255**, pp. 378–386
- [6] Lu T.K., Khalil A.S., Collins J.J.: 'Next-generation synthetic gene networks', *Nat. Biotechnol.*, 2009, **27**, pp. 1139–1150
- [7] Harris A.W., Dolan J.A., Kelly C.L., *ET AL.*: 'Designing genetic feedback controllers', *IEEE Trans. Biomed. Circuits Syst.*, 2015, **9**, pp. 475–484
- [8] Gardner T.S., Cantor C.R., Collins J.J.: 'Construction of a genetic toggle switch in *Escherichia coli*', *Nature*, 2000, **403**, pp. 339–342
- [9] Novak B., Tyson J.J.: 'Design principle of biochemical oscillator', *Nat. Rev. Mol. Cell Biol.*, 2008, **9**, pp. 981–991
- [10] Purcell O., Savary N.J., Grierson C.S., *ET AL.*: 'A comparative analysis of synthetic genetic oscillators', *J. R. Soc. Interface*, 2010, **7**, (52), pp. 1503–1524
- [11] O'Brien E.L., Itallie E.V., Bennett M.R.: 'Modeling synthetic gene oscillator', *Math. Biosci.*, 2012, **236**, (1), pp. 1–15
- [12] Teo J.Y., Woo S.S., Rahul Sarpeshkar R.: 'Synthetic biology: a unifying view and review using analog circuits', *IEEE Trans. Biomed. Circuits Syst.*, 2015, **9**, pp. 453–487
- [13] Wang R., Chen L., Aihara K.: 'Construction of genetic oscillators with interlocked feedback networks', *J. Theor. Biol.*, 2006, **242**, pp. 454–463
- [14] Gonze D.: 'Modeling the effect of cell division on genetic oscillators', *J. Theor. Biol.*, 2013, **325**, pp. 22–33
- [15] Elowitz M.B., Leibler S.: 'A synthetic oscillatory network of transcriptional regulators', *Nature*, 2000, **403**, pp. 335–338
- [16] Friedland A.E., Lu T.K., Wang X., *ET AL.*: 'Synthetic gene networks that count', *Science*, 2009, **324**, pp. 1199–1202
- [17] Buchler N.E., Gerland U., Hwa T.: 'On schemes of combinatorial transcription logic', *Proc. Natl. Acad. Sci. U.S.A.*, 2003, **100**, (9), pp. 5136–5141
- [18] Silva-Rocha R., de Lorenzo V.: 'Mining logic gates in prokaryotic transcriptional regulation networks', *FEBS Lett.*, 2008, **582**, (8), pp. 1237–1244
- [19] Buchler N.E., Gerland U., Hwa T.: 'Nonlinear protein degradation and the function of genetic circuits', *Proc. Natl. Acad. Sci. U.S.A.*, 2005, **102**, (27), pp. 9559–9564
- [20] Tamsir A., Tabor J.J., Voigt C.A.: 'Robust multicellular computing using genetically encoded NOR gates and chemical wires', *Nature*, 2011, **469**, pp. 212–215
- [21] Goñi-Moreno A., Amos M.: 'A reconfigurable NAND/NOR genetic logic gate', *BMC Syst. Biol.*, 2012, **6**, pp. 1–11
- [22] Sayut D.J., Niu Y., Sun L.: 'Construction and enhancement of a minimal genetic AND logic gate', *Appl. Environ. Microbiol.*, 2009, **75**, (3), pp. 637–642
- [23] Terzer M., Jovanovic M., Choutko A., *ET AL.*: 'Design of a biological half adder', *IET Synth. Biol.*, 2007, **1**, pp. 53–58
- [24] Zabet N.R., Hone A.N.W., Chu D.F.: 'Design principles of transcriptional logic circuits'. Proc. Alife XII Conf., Odense, 2010, pp. 186–193
- [25] Hoteit I., Kharna N., Varin L.: 'Computational simulation of a gene regulatory network implementing an extendable synchronous single-input delay flip-flop', *BioSystems*, 2012, **109**, pp. 57–71
- [26] Subsoontorn P., Endy D.: 'Design and analysis of genetically encoded counter', *Procedia Comput. Sci.*, 2012, **11**, pp. 43–54
- [27] Chuang C.H., Lin C.L., Chang Y.C., *ET AL.*: 'Design of synthetic biological logic circuits based on evolutionary algorithm', *IET Syst. Biol.*, 2013, **7**, (4), pp. 89–105
- [28] Chuang C.H., Lin C.L.: 'Synthesizing genetic sequential logic circuit with clock pulse generator', *BMC Syst. Biol.*, 2014, **8**, pp. 1–15
- [29] Nielsen A.A.K., Der B.S., Shin J., *ET AL.*: 'Genetic circuit design automation', *Science*, 2016, **352**, (6281), aac7341
- [30] Goldman N., Bertone P., Chen S., *ET AL.*: 'Towards practical, high-capacity, low-maintenance information storage in synthesized DNA', *Nature*, 2013, **494**, pp. 77–80
- [31] Church G.M., Gao Y., Kosuri S.: 'Next-generation digital information storage in DNA', *Science*, 2012, **337**, (6102), p. 1628
- [32] Lin C.L., Hu Y.J.: 'On the design of static genetic memory'. Proc. IEEE Industrial Electronics and Applications Conf., Hefei, 2016, pp. 1891–1896
- [33] Chen P.K., Lin C.L.: 'Synthesis of genetic clock with combinational biologic circuits', *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2015, **12**, (5), pp. 1206–1212