

Diseño y simulación de un medio sumador con desplazamiento de cadenas de ADN

Ana Karen Velázquez Sánchez and Israel Marck Martínez Pérez *

Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Baja California, México
anelvaz@cicese.edu.mx, israelmp@cicese.mx

Resumen Uno de los paradigmas más importantes en el área de Nanotecnología con ADN son las reacciones de desplazamiento de cadenas. Dicha técnica permite la implementación de dispositivos computacionales sin la necesidad de componentes adicionales (e.g. enzimas), lo que admite programar el cálculo únicamente en términos de secuencias de nucleótidos. En este trabajo se presenta el diseño y simulación de un circuito medio sumador utilizando compuertas lógicas (XOR, AND) implementadas a partir de este tipo de reacciones. Las estructuras se diseñaron utilizando el lenguaje DSD y se simularon estocásticamente mediante la herramienta Visual DSD. A diferencia de enfoques anteriores, la función OR exclusivo se emula directamente como una sola compuerta, en lugar de ser construida por medio de otras compuertas básicas. Los resultados computacionales mostraron una fiabilidad por encima de 98 % para todos los casos analizados.

Keywords: Cómputo Molecular, Biocomputación, desplazamiento de cadenas de ADN, compuertas lógicas, XOR, AND, medio sumador.

1. Introducción

El ADN es la molécula que almacena la información genética de los seres vivos y éste se utiliza como un medio para la computación molecular [1]. Los dispositivos moleculares construidos usando este material son útiles para aplicaciones en diversas áreas, tales como bio-detección, biomimética molecular y administración de fármacos. Sin embargo, el diseño de mecanismos de ADN correctos y robustos implica un gran desafío, ya que existe la posibilidad de que se incluya interferencia no deseada entre las moléculas del sistema [2]. Diversas estrategias han sido propuestas para implementar circuitos de ADN. Algunas de ellas dependen de agentes externos tales como enzimas de restricción para funcionar [3]. En comparación con estos enfoques, el desplazamiento de cadenas es una alternativa que tiene la ventaja de requerir solamente complejos de cadenas sencillas de ADN (que son relativamente baratos y fáciles de conseguir) sin la

* Esta investigación fue apoyada por subvenciones de Consejo Nacional de Ciencia y Tecnología(CONACyT) SEP-CONACYT-CB-2010-01-154863 para IMMP y beca CONACyT 275672-340234 para AKVS.

necesidad de que intervengan otro tipo de moléculas. Esta técnica se basa únicamente en la hibridación entre las secuencias de nucleótidos complementarias para llevar a cabo los pasos de cálculo [4]. Los circuitos construidos por medio de este enfoque son eficientes en cuanto a tiempo y energía [5] además de presentar una amplia gama de aplicaciones tales como terapéutica médica *in vivo*, instrumentos moleculares *in situ* y diagnóstico biomédico *in vitro* [2].

Una reacción de desplazamiento de cadenas se describe como el proceso mediante el cual dos hebras con complementariedad parcial o completa hibridan entre sí, desplazando en el proceso una o más hebras pre-hibridadas (Figura 1(a)). En los diseños de desplazamiento de cadenas, las secuencias se dividen en segmentos llamados dominios, los cuales se representan por medio de un número o letra y su parte complementaria con el mismo símbolo acompañado de un asterisco. La cinética del desplazamiento de cadenas se controla mediante ciertos mecanismos llamados puntos de apoyo (o en inglés *toeholds*), que no son más que secuencias cortas de ADN encargadas de iniciar la reacción. El desplazamiento de cadenas mediado por *toeholds* requiere la adición de cadenas sencillas externas para activar el proceso. Estas reacciones son programables gracias al diseño meticuloso de las secuencias que se utilizan; esto comúnmente implica el uso de *toeholds* cortos (aproximadamente 6 nucleótidos) y secuencias largas para los dominios a desplazar (alrededor de 20 nucleótidos) [6]. En una reacción en cascada, la salida de una reacción previa sirve como entrada para la siguiente reacción. Esto elimina la necesidad de añadir cadenas adicionales para desencadenar las reacciones en cada paso, lo que posibilita la construcción de sistemas autónomos complejos. En particular, el diseño de circuitos lógicos moleculares es relevante para la fabricación de nanodispositivos con funcionalidades cada vez más robustas, como por ejemplo, la realización de operaciones aritméticas [7].

En este trabajo se presenta un modelo de circuito medio sumador utilizando compuertas lógicas XOR y AND implementadas a partir de reacciones de desplazamiento de cadenas. A diferencia de otros enfoques, la función OR exclusivo se emula directamente como una sola compuerta, en lugar de ser construida por medio de otras compuertas básicas. Se presentan también resultados de simulaciones estocásticas que validan nuestro modelo. El resto del artículo se organiza como sigue. La siguiente sección describe de manera resumida los principales modelos de circuitos lógicos con desplazamiento de cadenas, haciendo énfasis en aquellos que abordan el diseño de sumadores. En la Sección 3 se explican las convenciones de los modelos propuestos. Posteriormente, se presentan los detalles de las simulaciones computacionales y resultados (Sección 4). Por último, se discuten las conclusiones y el trabajo a futuro.

2. Estado del arte

Recientemente, se han utilizado las reacciones de desplazamiento de cadenas de ADN para el desarrollo de circuitos lógicos [8]. Un ejemplo de esto se puede observar en [9] donde se presenta un diseño modular y la implementación experimental de compuertas AND, OR y NOT así como la construcción de circuitos de

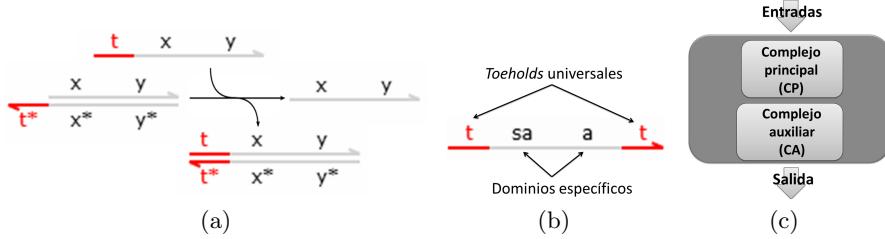


Figura 1. Convenciones de diseño: (a) ejemplo de reacción de desplazamiento de cadenas, (b) representación de una variable ‘A’ y (c) esquema de compuerta.

restauración, amplificación y regeneración de señales. El mayor inconveniente de este trabajo es la enorme cantidad de tiempo requerido para generar resultados, ya que el circuito con restauración de señal tarda por lo menos 10 horas en activarse. En [5] se presenta un diseño teórico para implementar cualquier circuito booleano usando el mecanismo de desplazamiento de cadenas. Una desventaja es que puede existir diafonía entre los buffers si no se tiene cuidado con el diseño de los *toeholds*. Por otro lado, en [2] se reporta la construcción de un circuito que calcula el piso de la raíz cuadrada de un número binario de cuatro bits. Dicho circuito está compuesto básicamente de compuertas AND, OR, NOT, NAND y NOR. La construcción del circuito se llevó a cabo con un mecanismo de reacción simple basado en el proceso de desplazamiento de cadenas reversible. En ese mismo año, Qian y Winfree hacen otra contribución al área utilizando bloques modulares para la construcción de los circuitos [10]. Cada bloque consiste en una compuerta sencilla que está configurada de tal manera que una entrada convierta catalíticamente el combustible en una salida si la concentración de la entrada excede el nivel de umbral. Aunque los autores sugieren que dicha propuesta puede ser escalable para circuitos de miles de compuertas, para dicha implementación se necesitaría generar más secuencias diferentes, además de que los circuitos resultarían bastante lentos y no garantizarían resultados correctos.

A la fecha se han propuesto algunos modelos de circuito medio sumador [7,11,12]. En el primer enfoque [7], el circuito fue implementado en laboratorio utilizando técnicas de desplazamiento de cadenas para la formación y disociación de una estructura G4-ADN. El cambio estructural representa las salidas del circuito y se indican con fluorescencia. Más recientemente, se reportó la construcción experimental de un medio sumador utilizando compuertas lógicas AND y YES/OR inspiradas en la compuerta INHIBIT [11]. Estas compuertas se basan en desplazamiento de cadenas y se controlan mediante nanopartículas de oro. Por último, en [12] se presenta un diseño conceptual (no simulado) de un medio sumador con desplazamiento de cadenas localizado. El proceso consiste en configurar el circuito lógico utilizando un tablero de origami de ADN. Este modelo solamente utiliza compuertas AND diseñadas a partir de estructuras secundarias de lazo.

Además, existen algunos trabajos que reportan simulaciones de sumadores completos basados en compuertas reversibles *seesaw* [13,14,15]. Estas compuertas tienen 3 etapas: *i*) liberación de señal encargada del proceso de reversibilidad; *ii*) una reacción de umbral que se procesa de manera simultánea con la anterior; *iii*) lectura de la salida por medio de fluorescencia. Cabe mencionar que la implementación de una compuerta lógica con este modelo requiere al menos de 3 compuertas *seesaw*. Dado que el diseño original no cuenta con XOR, los circuitos sumadores están formado por compuertas AND y OR. De esta manera, como se reporta en [15], se requieren al menos 46 estructuras *seesaw* para lograr el funcionamiento de un sumador completo.

3. Diseño conceptual del modelo

En el diseño propuesto, las variables de entrada y salida presentan el mismo formato y se representan como cadenas sencillas de ADN (orientación 5' – 3'). Su valor booleano está dado por la ausencia ('0') o presencia de éstas ('1') en alguna concentración. En la Figura 1(b) se puede apreciar la representación esquemática de una variable (A), compuesta de dos *toeholds* universales (uno en cada extremo) y dos dominios largos específicos. Éstos últimos deben ser distintos para cada una de las variables en el sistema ya que esto permite su diferenciación.

Cada modelo de compuerta está compuesto por dos complejos de ADN: el complejo principal y el auxiliar. Tal como indica el esquema de la Figura 1(c), las variables de entrada reaccionan con el complejo principal (CP) para liberar una *cadena intermedia* (CI) que interactúa con el complejo auxiliar (CA), el cual libera o atrapa la variable de salida según sea el caso. Por ejemplo, en el caso de la compuerta AND (Figura 2(a)), solamente se libera la cadena de salida si ambas entradas están presentes ('1'): la cadena intermedia se separa del complejo principal e interactúa con el complejo auxiliar, liberando la cadena de salida *S*. El OR exclusivo (XOR) (Figura 2(b)), también denominado anticoincidencia y comparador, regresa una salida '1' cuando solamente una de sus entradas es '1'; en todos los demás casos la salida permanece en '0'. Las reacciones para liberar la cadena de salida *C* ('1') ocurren en el complejo principal por medio de cadenas intermedias (Figura 3). Es importante mencionar que en el caso ($A=1, B=1$), cuando la salida corresponde a '0', se libera una CI desde el complejo principal, la cual reacciona con el auxiliar para dejar expuesto un *toehold* complementario. Dicha región atrapa la variable de salida *C*, indicando un '0' lógico.

Con las compuertas anteriores se puede emular el comportamiento de un circuito medio sumador (Figura 4(a)). Este circuito suma dos bits y produce como salida la suma de ambos (*S*) y el acarreo (*C*), según su tabla de verdad (Figura 4(b)). El acarreo (*C*) representa un desbordamiento en el siguiente dígito y su valor es '1' sólo cuando A y B tienen el valor de '1'; por tanto, es una operación AND entre A y B. La suma (*S*) es '1' si las variables A y B son distintas, por lo que puede expresarse como una operación OR – Exclusiva entre las entradas [16].

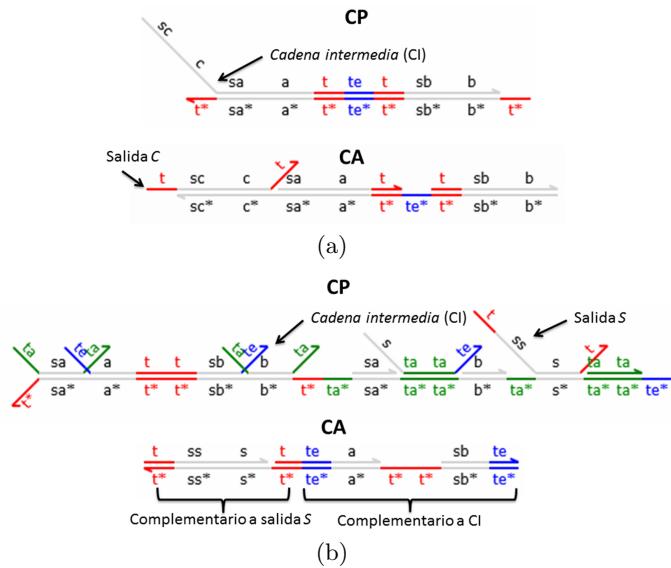


Figura 2. Modelos de compuertas: (a) AND y (b) XOR.

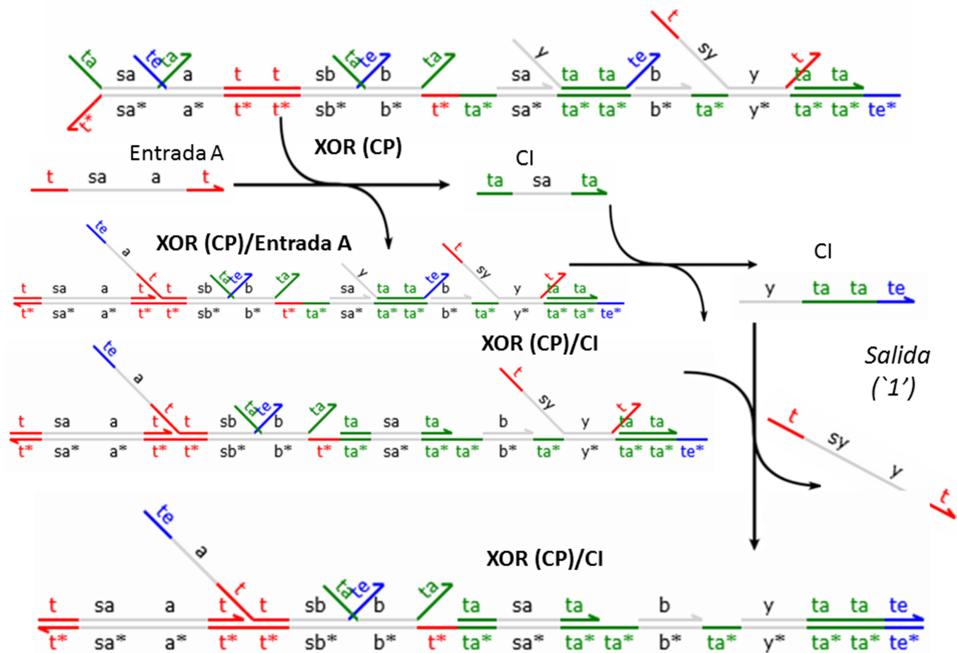


Figura 3. Reacción esquemática del caso ($A=1$, $B=0$) de la compuerta XOR.

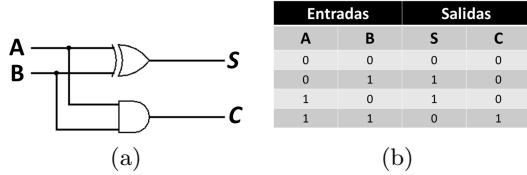


Figura 4. Medio sumador: (a) circuito lógico y (b) tabla de verdad.

4. Simulaciones computacionales y resultados

Las simulaciones se realizaron con el algoritmo de Gillespie [17], mediante la herramienta Visual DSD [18]. Este software es una aplicación de lenguaje DSD [19]. Cada simulación consistió en 30 ejecuciones para cada uno de los casos de la tabla de verdad del medio sumador. Los parámetros utilizados fueron los establecidos por defecto en el software, los cuales a su vez corresponden a los reportados en la literatura:

1. Velocidad de migración de ramificaciones de 8000 s^{-1} por nucleótido. La tasa de migración de ramificaciones para un dominio de longitud L está dado por r/L^2 , donde r es la tasa de migración por nucleótido [4].
2. Tasa de hibridación de $3 \times 10^{-4} nM^{-1}s^{-1}$. Se refiere a la constante de hibridación de los *toeholds*, y cuyo valor es el predeterminado por la herramienta Visual DSD [19].
3. Tasa de desnaturalización de 0.1126 s^{-1} . Corresponde a la constante de desnaturalización de los *toeholds* utilizada por defecto en el software Visual DSD [19].
4. 1000 puntos de muestra. La salida del simulador está dada por poblaciones de especies de ADN en puntos discretos del tiempo de simulación, por lo que este parámetro indica cuántas veces se registra el estado del sistema.
5. Tiempo de simulación de 1,500 segundos. El tiempo fue definido de manera empírica de acuerdo con la convergencia observada del sistema.
6. Número de moléculas. Se utilizaron 100,000 moléculas para los complejos auxiliares y principales y 200,000 moléculas para las entradas (dado que las compuertas AND y XOR reciben las mismas entradas). Este parámetro fue tomado del artículo [20].
7. Secuencias de nucleótidos predeterminadas por visual DSD. El banco de secuencias fue obtenido de [21] para *toeholds* y de [10] para dominios largos.

Tal como se puede observar en la Tabla 1, la fiabilidad de ambas compuertas lógicas (AND, XOR) en el circuito fue igual o mayor a 98 % para todos los casos. Los resultados experimentales se muestran en la Figura 5, donde se puede observar el promedio de las simulaciones. La Figura 5(a) muestra el caso ($A=0$, $B=0$) donde se obtuvieron 0 moléculas de S y C , indicando un 100 % de fiabilidad. En el caso ($A=0$, $B=1$) (Figura 5(b)), obtuvimos un promedio de 99,958 moléculas y 0 moléculas de C , sugiriendo una fiabilidad de 99.9 % y 100 %. Para

Tabla 1. Tabla con promedios de concentraciones finales de las 30 simulaciones.

Caso de la tabla de verdad	Moléculas			
	Salida S	Eficiencia S	Salida C	Eficiencia C
Sin entradas (0,0)				
Promedio	0	100 %	0	100 %
Desviación estándar	0		0	
Entrada B (0,1)	Salida S	Eficiencia S	Salida C	Eficiencia C
Promedio	99,958	99.9 %	0	100 %
Desviación estándar	13.464		0	
Entrada A (1,0)	Salida S	Eficiencia S	Salida C	Eficiencia C
Promedio	99,957	99.9 %	0	100 %
Desviación estándar	4.999		0	
Ambas entradas (1,1)	Salida S	Eficiencia S	Salida C	Eficiencia C
Promedio	228	99.7 %	98,682	98.6 %
Desviación estándar	9.438		39.007	

el caso ($A=1, B=0$) (Figura 5(c)), la concentración final promedio fue de 99,957 moléculas de S y 0 de C , por lo que también reportamos fiabilidades de 99.9 % y 100 %, respectivamente. Para el último caso de la tabla de verdad ($A=1, B=1$) (Figura 5(d)), en promedio se obtuvieron 228 moléculas de S indicando una tasa de error del 0.22 % y 98,682 moléculas de C , lo que nos da una fiabilidad del 98.6 %. En cuestión de tiempo de cálculo es importante mencionar que las simulaciones convergen a los 1,500 segundos o antes, tal como se puede observar en las gráficas.

5. Conclusión

Las resultados indican que las reacciones ocurrieron de la forma esperada y que el medio sumador pudo construirse con los modelos propuestos de manera satisfactoria. Además, las compuertas lógicas resultaron ser eficientes y de rápida respuesta (25 minutos o menos) tal como sugieren los experimentos computacionales realizados. A pesar de que los resultados muestran una alta eficiencia, todas las simulaciones reportaron el porcentaje más bajo (98 %) en el caso donde ambas variables están presentes ($A=1, B=1$). Intuitivamente, esto podría resolverse mediante un incremento en el tiempo de simulación o en la concentración de moléculas. Como trabajo futuro, sería recomendable realizar un análisis detallado sobre la sensibilidad de los parámetros y su impacto en el desempeño del circuito. En particular, resultaría de interés estudiar aquellos parámetros que tienen relación directa con la longitud de las secuencias, como la velocidad de migración y la tasa de desnaturalización. De igual manera, se pretende escalar el modelo propuesto a un sumador completo y verificar su robustez con el conjunto de parámetros obtenido. Por otro lado, sería también interesante comparar la eficiencia de dicho sumador completo con respecto a otros resultados reportados

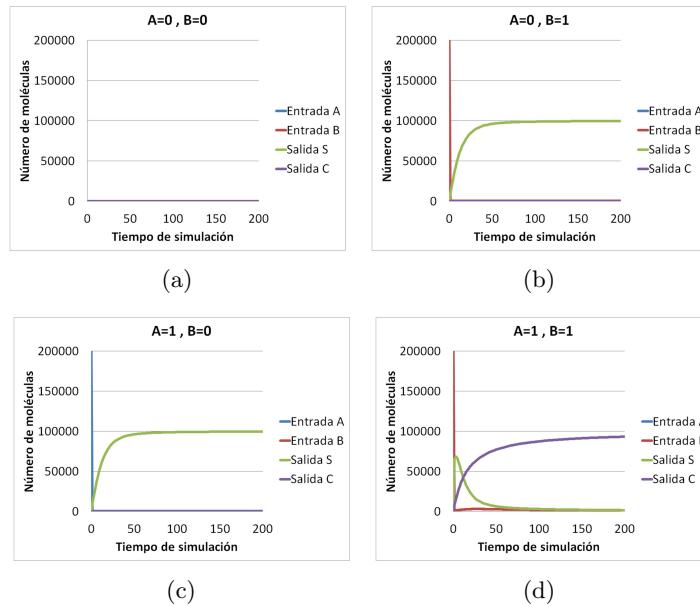


Figura 5. Simulación de circuito medio sumador con 100,000 moléculas de entrada para compuertas y 200,000 moléculas para las entradas a 200 s: (a) caso 1 ($A = 0, B = 0$); (b) caso 2 ($A = 0, B = 1$); (c) caso 3 ($A = 1, B = 0$); (d) caso 4 ($A = 1, B = 1$).

en la literatura (los circuitos medio sumador reportados anteriormente no incluyen pruebas de simulación). Por último, dado que las compuertas diseñadas son funcionalmente limitadas, otra línea de investigación consiste en complementar nuestro modelo con el diseño de otros componentes (e.g. compuertas NOT y OR) con la idea de poder implementar cualquier función Booleana.

Referencias

1. Adleman, L.M.: Computing with DNA. *Scientific american* **279**(8) (1998) 34–41
2. Qian, L., Winfree, E.: Scaling up digital circuit computation with DNA strand displacement cascades. *Science (New York, N.Y.)* **332**(6034) (June 2011) 1196–1201
3. Ran, T., Kaplan, S., Shapiro, E.: Molecular implementation of simple logic programs. *Nature Nanotechnology* **4**(10) (2009) 642–648
4. Lakin, M.R., Youssef, S., Cardelli, L., Phillips, A.: Abstractions for DNA circuit design. *Journal of The Royal Society Interface* **9**(68) (2012) 470–486
5. Chiniforooshan, E., Doty, D., Kari, L., Seki, S.: Scalable, time-responsive, digital, energy-efficient molecular circuits using DNA strand displacement. In: *DNA computing and molecular programming*. Springer (2011) 25–36
6. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences* **107**(12) (2010) 5393–5398

7. Xu, S., Li, H., Miao, Y., Liu, Y., Wang, E.: Implementation of half adder and half subtractor with a simple and universal DNA-based platform. *NPG Asia Mater* **5** (Dec 2013) e76
8. Zhang, D.Y., Seelig, G.: Dynamic DNA nanotechnology using strand-displacement reactions. *Nature chemistry* **3**(2) (2011) 103–113
9. Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E.: Enzyme-free nucleic acid logic circuits. *Science* **314**(5805) (2006) 1585–1588
10. Qian, L., Winfree, E.: A simple DNA gate motif for synthesizing large-scale circuits. In: *DNA computing*. Springer (2009) 70–89
11. Wu, Z., Dong, B., Zhou, X., Shen, A., Hu, J.: Inhibit-inspired two-output DNA logic gates based on surface-enhanced raman scattering. *Chemistry – A European Journal* (2015)
12. Muscat, R.A., Strauss, K., Ceze, L., Seelig, G.: DNA-based molecular architecture with spatially localized components. 40th International Symposium on Computer Architecture **41**(3) (June 2013) 177–188
13. Xiangrong, L., Juan, S., Juan, L., Yan, G., Xiangxiang, Z.: Molecular logic computation with debugging method. *Journal of Nanomaterials* **2015** (2015) 11
14. Wang, Z., Tian, G., Wang, Y., Wang, Y., Cui, G.: Multi-digit logic operation using DNA strand displacement. In Pan, L., Păun, G., Pérez-Jiménez, M., Song, T., eds.: *Bio-Inspired Computing - Theories and Applications*. Volume 472 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg (2014) 463–467
15. Wang, Y., Tian, G., Hou, H., Ye, M., Cui, G.: Simple logic computation based on the DNA strand displacement. *Journal of Computational and Theoretical Nanoscience* **11**(9) (2014-09-01T00:00:00) 1975–1982
16. Tocci, R.J., Widmer, N.S.: *Sistemas digitales: principios y aplicaciones*. Pearson Educación (2003)
17. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81**(25) (1977) 2340–2361
18. Lakin, M., Youssef, S., Polo, F., Emmott, S., Phillips, A.: Visual DNA: a design and analysis tool for dna strand displacement systems. *Bioinformatics* **27**(22) (November 2011) 3211–3213
19. Phillips, A., Cardelli, L.: A programming language for composable DNA circuits. *Journal of the Royal Society Interface* **6**(S4) (August 2009) 419–436
20. Martínez-Pérez, I.M., Zimmermann, K.H., Ignatova, Z.: An autonomous dna model for finite state automata. *Int. J. Bioinformatics Res. Appl.* **5**(1) (January 2009) 81–96
21. Zhang, D.Y., Turberfield, A.J., Yurke, B., Winfree, E.: Engineering entropy-driven reactions and networks catalyzed by DNA. *Science* **318**(5853) (2007) 1121–1125