RTL2DNA: an Automatic Flow of Large-Scale DNA-based Logic Circuit Design

Zohre Beiki, Ali Jahanian*

Received: date / Accepted: date

Abstract DNA computing is a new kind of computation for solving the complex problems with the huge degree of parallelism. Recently it is found that DNA-based logic systems can be useful in many of biomedical applications such as early cancer detection. DNA logic systems have been applied successfully to detect the risky patterns of nucleotide-based cancer biomarkers (microRNAs). Detection of real diseases requires the large-scale DNA-based logical systems. Therefore, large-scale DNA-based logic circuits is a crucial research topic.

In this paper, an automatic design flow is proposed to facilitate the design, verification and physical implementation of multi-stage and large-scale DNA logic circuits. Digital Microfluidic Biochips (DMFB) have been used recently as a promising platform for efficient implementation of DNA-based computing systems and circuits. We used this technology as the physical platform for implementation of DNA-based circuits.

Our experiments and implementations show the feasibility, accuracy, efficiency and simplicity of the proposed design flow. Final DNA reactions that are synthesized by the proposed design flow are verified and simulated with stochastic DNA-reaction simulators to prove the correctness of the proposed design flow. This design flow can open a new horizon for researchers and scientists to design, implement and evaluate the DNA-based logic systems.

Keywords: DNA circuit, DNA computer, Micro-architecture, DNA design flow, Automatic Design Flow.

Z. Beiki
Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.
Tel: +98-31-3793-4057
E-mail: z.beiki@eng.ui.ac.ir

A. Jahanian
Faculty of Computer Science and Engineering, Shahid Beheshti University, G. C., Velenjak, Tehran 19839-63113, Iran.
Tel: +98-21-2990-4188
Mobile number: +98-9126541308
E-mail: jahanian@sbu.ac.ir
many years ago DNA molecules were known as protein producers and transfer the genetic features to the next generations, but in recent decades, it is found that DNA molecules can be used for a new type of computation [1]. Leonard Adleman appropriated the DNA molecules to solve the Hamiltonian path problem [2]. Adleman’s experiments show that NP-complement problems can be solved using DNA molecules while this could not be done with silicon-based computers, thanks to the significantly huge degree of parallelism. DNA computation is a bridge between biological science and computer engineering. some applications of DNA computing are including of gene analysis, medical therapeutics, pharmacy, and solving NP-Complete/NP-Hard problems [3] and [4].

bio-computing especially DNA computing and DNA nanostructures design methods have been developed over the last two decades to realize and control matter at the Nanoscale [5]. The following provided review of the previous work on DNA-based logic gate design.

the concept of localized DNA strand displacement was firstly proposed by Sakamoto et al., this proposed method is a mechanism for implementing the chemical reaction network on a surface of a DNA Nanostructure [6]. In [7] the authors proposed a method of AND/OR functions implementation based on DNA strands. their proposed structure, consisting of micro-reactors along with attached heating elements towards controlling the DNA annealing process. this structure can be used to solve the satisfiability problem in linear space and quadratic time [7]. DNA localized circuits proposed by Qian in 2014, Localization accelerate the kinetics by increasing the relative concentration of strand reaction [8]. Fan et al. in [9] proposed a three-input label-free/enzyme-free majority gate through DNA hybridization without DNA replacement and enzyme catalysis; further, the system is capable of implementing various basic/cascade logic gates.

The Seesaw logic model was a breakthrough for circuit design based on DNA strands. This model was proposed by Qian et al. in 2009 [10] and further improved in 2011 [11]. Their methodology increased scalability and reliability in DNA circuits compared with earlier methods however this method increased the number of strands.

Recently, DNA logic systems have been utilized successfully to detect the microRNAs, microRNAs are biomarkers based on nucleotides that their concentration changes in different types of cancer [12] and [13]. Hemphill and Deiters applied the DNA logic gates to create a molecular system with MicroRNAs inputs for cancer detection [12]. DNA circuits that recognize microRNA cancer biomarkers through strand hybridization.

Microfluidic biochips are controlled and automated platforms that are used for chemical reactions [14]. These chips are known as a promising platform for executing DNA operations in a controlled process. In [15] discusses a flexible configuration platform for performing a DNA computation on a microfluidic architecture in order to realize basic logic structures such as switches,
ries, and logic gates; Their proposed design is capable of programming DNA strands into various Boolean problems. However, each technique comes with its own advantages and disadvantages.

More than 40 years electronic design experiences show that Computer-Aided Design (CAD) plays a fundamental role in progress of the VLSI. Really, the benefits and drawbacks of an emerging technology cannot be evaluated without a suitable CAD tool. The progress of DNA computing will be boosted when there is an efficient CAD environment that enables the researcher to implement and evaluate their ideas on this platform [16]. Some very limited design flow are proposed in recent years for simulation of DNA reactions. Chris Dwyer in [16] presented a scientific tool for DNA self-assembly design that enables the design of SSI and MSI systems with DNA strands. Circuit design with DNA self-assembly associated with many of the challenges contains costs and yield, [16]. DNA self-assembly circuits are not expandable and often they are used as a base to build other structures [17], [18]. Another design tool proposed by Selmihin et al in [19] for DNA origami structures. They provided a small tool for designing simple circuits such as a simple DNA origami biosensor device. A placement algorithm is proposed in [20] for localized DNA logic circuits. In [21] seesaw compiler toolbox is used to convert the AND/OR circuits to dual-rail seesaw logic circuits. Output of seesaw compiler can be simulated and evaluated using the standard Visual-DSD simulator.

DENA architecture is A configurable DNA Architecture based on microfluidic biochips that are used for the implementation of DNA large-scale circuits this proposed method is introduced in [22]. DENA can improve the cascade-ability and feasibility of DNA circuits in addition, the basic concepts of configurable DNA architectures are described in this work. However, this paper does not address any automatic design flow for this technology.

As mentioned before, considerable contributions have been addressed on design and analysis of DNA-based logic gates and circuits for computing and medical applications. However, no scalable and easy-to-use design flow is introduced for the DNA-based logic systems. This problem makes the design and evaluation of DNA-based systems difficult and even infeasible for researchers and designers of this scope.

This paper proposed a scalable and automatic design flow for DNA circuits that enables synthesis, simulation and implementation of DNA logic systems automatically. This design flow called RTL2DNA in this paper. RTL2DNA flow provides a feasible and straightforward design flow from RTL to physical implementation of combinational logic circuits based on DNA strands. MFBC are used in this paper as the infrastructure of DNA logic gates. In other words, the proposed design flow implements the DNA logic gates on MFBCs. The main contributions of this paper as follows:

- providing an RTL-to-DNA simulation/synthesis flow for large-scale DNA-based logic circuits that enable the design, implementation, and verification of large DNA circuits. This feature enables the researchers to implement and evaluate their ideas in this area easily.
using the microfluidic biochips as the implementation platform provides a controlled framework for realizing the DNA-based logic circuits. The proposed design flow synthesized an RTL logic circuit into the biochemical assay that can be executed on digital microfluidic biochips.

we used the microarchitecture of [22] as the base technology. A highlight feature of the used microarchitecture is that the number of required strands in the proposed method is constant for all the circuits and does not increase for the large circuits. This feature is a key contribution to the implementation of large Boolean systems based on DNA logic gates.

The rest of this paper is organized as follows. Section 2 provides a brief review of DNA-based logic design styles and Section 3 reviews the digital microfluidic technology and its application for DNA computing. Section 4 explains the used DNA micro-architecture and the improvements that are made on it. The proposed design flow and the experimental results of RTL2DNA flow are described in Section 5 and 6, respectively and finally, Section 7 concludes the paper.

2 DNA-based Logic Circuit Design

The Watson-Crick complementary rules are the foundation of DNA-based computation. These rules definitions of reactions between DNA strands. Each DNA strand is included in a finite set of nucleotide acids. that are encoded by “Adenine” (A), “Guanine” (G), “Cytosine” (C) and “Thymine” (T) [23] and [24]. Proposed have been different methods for the design of DNA-based logic gates and there are mainly categorized into enzyme-based and enzyme-free categories. The enzyme-free category is recommended because of the lower costs, the speed of operation, and the simplicity of implementation. The enzyme-free method is called Toehold-mediated because these reactions used a small and fast DNA domine in the name of Toehold. this domain starts the reactions in the DNA displacement process. In this section, the concept of Toehold-mediated design and Seesaw DNA logic (as the most used method) are described briefly.

2.1 Toehold-mediated Logic Design

Toeholds start the reaction in the strand displacement method. in this method, we don’t have any enzyme compared with enzyme-based methods. Therefore strand displacement method is faster and cheaper. the most important disadvantage of the strand displacement method is the great number of orthogonal strands. By increasing the size of the circuit, the orthogonal strands Increase sharply.

In an article in 2000, Yurke et al. stated that the probability of binding two complementary DNA strands is dependent on the reverse of their length or the number of bases in strands. In simple terms, the shorter complement strands
have a higher merging probability.[25]. in fact, small strands in [25] are the Toehold strands that call Primer in biological sciences. In 2000, authors of [25] used Toehold attempted to accelerate the merging of DNA strands; Applying the Toehold strand improves the level of controllability of the DNA displacement process. Figure 1 reveal a simple example with of strand displacement method. Toehold strands are illustrated in red color, they are reactions starter strands. Strand displacement reactions continue using the After energy released of Toehold reaction. green strands represent a single DNA strand with its direction showing the merging direction (from 5 to 3). The labels of the complement strand are represented by the quote sign (*).

Figure 1 illustrates a strand displacement reaction, this reaction used Toehold strands for starting. 1-A show a complex strand consist of Toehole $C^*$ and double strand $B$ ($BB^*$). This complex strand will reacted by single upper strand $CB$. As shown in Figure 1-B, Initially, Toehold $C^*$ of complex strand bound to its complementary ($C$ in single strand $CB$). The energy released from this Toehold binding leads to continued reaction therefore, strand $B$ of $CB$ is replaced with strand $B$ in the initial complex strand (Figure 1-C). In the finally, reaction result as shown in Figure 1-C is consist of double strand $CB$ and single strand $B$.

The kinetic of Toehold strands depends on sequence nucleotides and the length of strands. The speed of strand displacement reactions regulates with the Toehold-mediated [26]. Typically, Toehold strands have lengths that range from 3 to 7 nucleotides.

2.2 Seesaw Logic Gate

Qian et al. proposed the Seesaw logic design technique in 2009 that is known as an efficient Toehold-mediated DNA logic design style [10]. A Seesaw logic gate is comprised of five strand types: inputs, outputs, gate, threshold and fuel in which the output strand is generated as the result of reactions between input, gate and threshold strands.

Seesaw gates apply two basic components based on Toehold-mediated method. The input strands react catalytically so that a single input strand is capable of releasing multiple output strands from several Seesaw gates. On the other hand, the output strands act as the input strands for the next Seesaw gate. A Seesaw-based system consisting of ‘n’ inputs and ‘m’ outputs incorporates ‘n’ input strands, 3 internal strands (gate, threshold and fuel strands) and m output strands.

Figure 2 shows the internal structure of an OR gate. This element can compute either (AND/OR) functions depending on the initial concentration of the Threshold DNA strand. Initial Threshold concentration is $600nM$ for an OR gate and the concentration of strands should be increased to $1200nM$ to generate an AND gate.
Input strands (Input1 and Input2) in Figure 2 react with Gate1 to produce upper strand $S_2.T.S_5$. This strand combined with Threshold strand that consists of two Toehold domain. So the concentration of $S_2.T.S_5$ can be controlled by Threshold concentration. Remaining $S_2.T.S_5$ reacts with strands Gate2 and $S_2.T.S_6$ and produces the output strand. It is worth noting that Fuel strand increases the chance of reactions to generate the output strand.

The major advantage of seesaw logic gates is the cascading capability of this design style since the output concentration of seesaw does not degrade considerably such that it can be used as the input of next stage of the system. However, there are some limitations in seesaw design methodology as follows:

– lacking of the inverter gate (NOT gate) complicates designing gates in the seesaw logic greatly. Dual-rail logic is used in seesaw to overcome this weakness in cost of considerable overheads in number of strands and system complexity.
– unintended reactions (i.e. crosstalk) are increased with the growing number of gates such that designing large circuits would be infeasible in practice.
– large number of orthogonal strands are required for medium- and large-size circuits such that designation of large circuits is very hard.

Seesaw design methodology is appropriate for SSI logic design with many of DNA strands but it is infeasible in case of larger circuit designs.

3 Digital Microfluidic and DNA Computing

A digital microfluidic biochip (DMFB) device is a platform for performing operations of biological assays in an automatic and controllable manner [27]. A DMFB can automatically manage droplets reaction on the two-dimensional silicon array. this work is done programming the electrodes under surface. Figure 3 illustrates structure of digital microfluidic biochips.

The core technology realizing DMFB devices is electro-wetting on dielectric (EWOD) [27]; the technology works based on applying an electrical field to the electrode beneath the droplet in order to actuate (move) the droplet [27]. Movement of droplets across the electrodes allows for fundamental microfluidic operations such as holding (storing), transporting (moving), merging, mixing and splitting. Further, the chips can accommodate other operations such as detecting, heating and cooling by the means of external equipment integrated into the chip during the manufacturing time.

3.1 Digital Microfluidic Biochips as the Platform for DNA computing

chemical reactions between the DNA strands are manual operations. It’s the most challenge of DNA computing. Recently, With the development of DMFB
automatically control of DNA computing reactions is hopeful and not far from the mind. [28]. Furthermore, microfluidic biochips can be utilized for providing scalability and flexibility demanded by large-scale DNA logic circuit designs [22].

The following items describe the proposed techniques for using the MFBCs in various steps of DNA computing:

- **Annealing:** Temperature ramp (annealing) is the most important operation in DNA computing. Authors of [29] shows that temperature changing is possible in the digital microfluidic biochip. They proposed a practical system for scalable and flexible mechanism to change the temperature on a MFBC for DNA strand displacement.

- **Strand separation:** Strand separation is required for separating of the output strands of a computation from environment to be used as the input strands of another DNA operation. Strand separation on the microfluidic biochip is performed in [30] with very good resolution.

- **Polymerase chain reaction:** Polymerase chain reaction (PCR) is an effective technique for producing the input strands of a DNA computer and for improving the concentration levels and strands reactions in DNA computing. Microfluidic platforms have been used successfully for performing the [28], [31], and [32].

As mentioned before, digital microfluidic biochips (DMFB) are a revolutionary solution for implementation of DNA-based computing system and circuits. We used DMFB as the controlling platform for executing the DNA operations.

### 4 DNA-based Micro-Architecture

In [22], a micro-architecture (DENA) has been proposed that can be used for large-scale circuits based on DNA computing. The proposed architecture design was inspired by FPGA and was configurable and useful for any automatic or semi-automatic design flow. Our design flow uses the revised version of DENA micro-architecture for automatic DNA-based circuit synthesis. DENA is regular architecture consists of the arbitrary number of DENA Clusters (DC) which can be configured to implement a 4-input logic function. Figure 4 shows the general structure of DENA micro-architecture with 2 × 2 DCs. As shown in this figure, each DC is implemented on a 3 × 3 grid of a microfluidic biochip.

Detailed structure and operation of this micro-architecture is described in [22]. We used the DENA [22] as the logical micro-architecture. We made some improvements on DENA to increase its capabilities for automatic synthesis. Rest of this section describes these improvements.
4.1 Improvements Applied to DLB Design

DLB is the basic functional element of a DENA cluster. Original DLB in [22] is implemented with 7 number of DNA-based configurable OR/AND gate (COA) that are shown in Figure 5-A. It implements a Sum-of-Product (SOP) or Product-of-Sum (SOP) with 8-input according to COA configuration. In this paper, DLB is upgraded from a 2-level AND/OR to 4-input lookup table that brings better performance/overhead tradeoff (Figure 5-B and 5-C).

As can be seen in Figure 5-C, modified DLB contains 16 number of 5-input AND gates. Therefore, 80 orthogonal DNA strands are required only for input of AND gates that are implemented with seesaw design method. On the other hand, DENA clusters contain 4-inputs as MUX select pins. Each DLB input is connected to different 8 number of AND gates and hence it should be converted to 8 orthogonal strands. We used conventional seesaw amplifying gate for generating the 8 orthogonal strands. Schematic view of amplifying gate is shown in Figure 6. More details about the amplifying gate can be found in [33].

We utilized 8 amplifying seesaw gates in this paper. Each amplifying gate received one input (A to D or $\overline{A}$ to $\overline{D}$) and produced 8 different strands. DLBs have 4 input strands, 1 output strand and 16 number of AND gates. The inputs of AND gates are different from each other. The threshold of AND gate is between $n-1$ and $n$ input concentration when inputs are ON. The threshold concentration for OR gate with n-inputs is between $n \cdot C_{off}$ and $1 \cdot C_{on}$ when $C_{off}$ and $C_{on}$ are the concentration for OFF and ON input mode ($100nM$ and $900nM$) [33]. AND/OR in DLB tile is similar to AND/OR seesaw gate that shown in figure 4 only by increasing the threshold concentration.

Each DLB is implemented with 16 AND gates, 5 number of OR gates (OR network) and one amplifying gate. Therefore, each DLB contains 320 different strands that 80 strands are used in amplifying gate and 4 strands are utilized for inputs.

4.2 Modifications Applied to the Inverter

As mentioned before, there is no straightforward solution for DNA-based Inverter design so that design of the inverter gate has been a serious challenge in DNA logic gate design. In [22], a 2-stage Inverter gate (NOT gate) is proposed. In this paper, we modified this NOT gate and used 4 numbers of NOT gates in each DENA Cluster. Each NOT gate is used for one input (A to D) and the output of the inverter is sent to DLB. In this way, DLB has 8 inputs (A to D and $\overline{A}$ to $\overline{D}$).

Moreover, each inverter (Step1 and Step2) have 2 different strands and we used 4 inverter gates. Therefore 8 different strands are used for inverter gates.
4.3 Improvements of the Converter

The output strands of a DLB should be used as input strands of the succeeding DLBs. A converter gate presented in [22] that reshapes the output of a DLB to form of the first input of next DLB. We revised the converter enabling transformation of the DLB output to an arbitrary input of the next DLB. This modification improves the logic utilization and flexibility of the design. An amplifying gate with $4$ output is used to design the Converter stage. DLB output sent to Converter stage (amplifying gate) then one of $4$ Converter output is sent to the next DC according to routing between DCs. Converter used $6$ new strands. Fuel concentration for this amplifying gate is $800nM$ and another concentration is similar to Figure 6.

5 The Proposed Design flow

The main contribution of this paper is introducing an automatic design flow for large-scale DNA circuits based on the improved version of DENA micro-architecture [22]. We revised the conventional electronic design flow and added some new steps to it for adapting to DNA-based logic design. Figure 7 shows the overall design process.

The suggested design cycle starts from an RTL description of the design. Berkeley ABC tool [34] is used as a technology-independent logic synthesis toolbox and TV-Pack [34] is modified to map the synthesized (output of ABC) netlist to DENA micro-architecture. After this mapping, design cycle is divided to $2$ sub-trees as follows.

- **simulation and verification:** Left-hand sub-tree of RTL2DNA in Figure 7 is simulation and verification. In the proposed flow, the synthesized netlist is converted to a sequence of reactions between the DNA strands that implements the circuit’s logic function. The DNA-based logic circuit can be simulated and verified by a stochastic DNA reactions simulator such as Microsoft Visual-DSD tool [35]. VisualDSD is a well-known DNA strand displacement modeling and simulation tool that is widely used in research related to DNA computing, especially for DNA logic gate design. It is worthwhile to note that the synthesized circuit is mapped to VisualDSD input model based on DENA microarchitecture.

- **physical design:** the right sub-tree of Figure 7 represents the physical design of the synthesized circuit on the microfluidic platform. Microfluidic biochips are responsible platform for running a chemical assay (a sequence of chemical operations). On the other hand, each DNA computation consists of a sequence of biochemical operations that can be encoded as an assay. In RTL2DNA design flow, synthesized logic description is mapped to a biochemical assay to be executed on a microfluidic biochip. We used UCR Static Synthesis Simulator (SSS) to implement the circuit on DMFB [36]. SSS is an open-source framework designed for supporting algorithmic and software-driven control for DMFBs. SSS toolbox [36] is revised
in this research to place and route the mapped function on DENA microarchitecture over the microfluidic biochip platform. Finally, a placed and routed circuit on microfluidic platform is generated that realizes the projected circuit on this platform.

The following subsections describe the components of RTL2DNA in more details.

5.1 Logic Design Flow

In this step, RTL description of the projected circuit (in Verilog) is synthesized into gate level and then is mapped to DENA microarchitecture. It is noting that Verilog-to-Routing (VTR) toolbox is a worldwide collaborative effort to provide an open-source framework for conducting FPGA architecture and CAD research and development [34]. We revised the TV-Pack (a component of VTR) for partitioning and mapping the design to DENA architecture. The mapped netlist, show the DENA Cluster (DC) index of each cluster, inputs and output of each DC and the connection between the DCs. Figure 8 represents the C17 circuit of ISCAS benchmark suite [37] in different design stages from RTL to DNA. Figure 8-A and Figure 8-B show RTL and corresponding gate-level description, respectively. The partitioned and mapped circuit that resulted by revised TV-Pack is shown in Figure 8-C that represents DC unique name and its index, configuration bitstream of the DC and its I/O connections.

At the end of this phase, the partitioned netlist is ready for physical design and DNA-based functional simulation. This netlist can be used not only for simulation, but also for the physical design of the synthesized circuit. The next subsections describe simulation and physical design process after the logic design.

5.2 Verification and Simulation

Verification of a DNA-based logic system is done by analyzing the reactions between input and gate DNA strands. We used VisualDSD toolbox for simulating the reactions between strands. At the first step, DSD file (input description file of VisualDSD) is generated for each synthesized DENA Cluster. It is worth noting that DSD files describe the different components of the DNA-based circuit that can be simulated with VisualDSD. At the second step, VisualDSD is used to simulate the DNA reactions between the DNA strands corresponding to the various parts of the synthesized circuit.

We proposed an algorithm to map the logical circuit to the DNA reactions. For this purpose, synthesized netlist should be converted to a DSD description. The generated DSD file is a DNA-based implementation of the synthesized circuit that can be simulated using VisualDSD. Figure 9 shows the presented
algorithm for Visual-DSD code generation. The following paragraphs describe the proposed algorithm in detail.

**Step 1:** at the first step, the description of un-configured DLB structures are generated based on seesaw logic style (i.e. "dlb.dna" file). In this file, each DLB is implemented in a DNA-based 4-input Seesaw lookup table.

**Step 2:** in this step, bitstream field of the final netlist is used to configure the lookup table. Minterms of DLB structure are configured by determining the concentration to the DNA strand corresponding of each minterm. At the end of this step, DLBs are configured by identifying the concentration of strands in the DSD description file.

**Step 3:** as mentioned before, each inverter includes two steps. Therefore, two different "not.dna" files are created. Concentrations of input strands are determined according to circuit primary inputs at simulation time.

**Step 4:** "conv.dna" file is also created based on this step. The concentration level of the converter is equal to DLB output concentrations. Therefore, it will be determined by the simulation of DLB. As an illustrative example, generated files and its configuration of DENA Cluster 0 for the IBM C17 benchmark [37] is shown in Figure 10. Due to the large size of files, only important parts of each file is shown. In DLB Configuration file (Figure 10-A, 3 minterms are ON based on the corresponding bitstream in Figure 8 (e.g. minterms 3, 7 and 11 in bitstream “0001000100010000”). This figure represents the VisualDSD files for inverter, converter and DLB (Figures 10-A to 10-D). Figure 10-E shows the simplified circuit for the specified DLB.

DENA has a matrix-like structure such that each DC in this architecture is connected to its neighboring DCs. Whole the DENA matrix cannot be simulated instantaneously because each DENA Cluster can be configured and simulated if and only if all DC inputs be available. In this scenario, concentration of the strands of each DC is configured and simulated based on the concentration level of its input strands. We generated a graph showing order of DCs in simulation process. This graph called as DENA Cluster Priority Graph (DCPG) in this paper. Figure 11 shows the DCPG for C17 benchmark.

Figure 11-A shows the proposed algorithm for generating the DCPG. This algorithm consists of two main steps. At the first phase, DENA clusters are classified as primary nodes, internal nodes and output nodes and then DCPG is created based on this classification at the second phase.

After creating the DCPG, primary input nodes should be processed at the first time. Primary nodes are DENA clusters with only primary input. They do not depend to any other strands. Therefore, they can be simulated in parallel at the first step. All internal nodes must wait until their inputs have available. The output strands of the final nodes should be saved as primary output data. VisualDSD descriptive files simulate whole the circuit according to the DCPG. Now, each DC should be simulated based on the order specified by DCPG. In other words, an interactive simulation is scheduled such that level $i^{th}$ DLBs are simulated and their output strands are fed to level $i+1$ and then level $i+1$ is simulated.
As shown in Figure 12, DC inputs and their inverted signals are connected to DLB. DLB output is sent to the converter. Therefore, converter input concentration will be determined after the DLB simulation. The output of the converter can be sent to the next DC or primary output pins.

As mentioned before, we developed a toolchain to simulate the synthesized DNA-based logic systems. This toolchain converts the synthesized netlist to a sequence of DNA reactions that can be simulated using off-the-shelf DNA simulation tools.

5.3 Physical Design Flow

As mentioned before, a synthesized circuit is implemented on a digital microfluidic biochip (DMFB) in the RTL2DNA flow. In this approach, the synthesized circuit is converted to the standard assay format that can be implemented on a DMFB platform. We used UCR Static Synthesis Simulator (SSS) to implement the circuit on DMFB [36]. SSS is an open-source framework designed for supporting algorithmic and software-driven control for DMFBs.

The physical design process of SSS framework consists of three main phases; scheduling, placement, and routing. The SSS performs the scheduling algorithm based on the input of bioassay protocol and DMFB architecture specifications initially. This stage attempts to schedule microfluidic operations within the bioassay protocol given the available resources. Next, the placement algorithm executes and determines the location of scheduled microfluidic operations on the DMFB array of electrodes.

After that, the droplet routing algorithm is invoked to plan the moving pattern of droplets on the DMFB array of electrodes, either from input reservoirs to the modules, between the modules or from modules to the output reservoir. The output of the design cycle is a sequence of electrode activations that execute the assay on the DMFB. In addition, the UCR Static Synthesis Simulator offers various printed circuit board (PCB) wire-routing algorithms; which are aimed at reducing the number of control pins and PCB layers required to fabricate the chip. The framework also includes a suite of visualization tools for debugging and producing graphical output from scheduling, placement, droplet routing and wire-routing output files [36].

Figure 13 shows the inputs and outputs of the physical design flow for the C17 benchmark. These files generated automatically in the proposed flow as an input of SSS tool. Figure 13-A shows the generated assay from the synthesized circuit and Figure 13-B specifies details of the architecture file corresponding to the target microfluidic platform for SSS tool. Assay file describes the DFG of the operation on the microfluidic platform. Nodes of DFG can be four basic microfluidic operations (i.e. DISPENSE, MIX, SPLIT and OUTPUT). Architecture file determines the configuration of the microfluidic biochip. It specifies the input/output location, the time step of operation, drops frequency, location for the waste of useless strands and washing for microfluidics.
The RTL2DNA flow enables the automatic design and verification of the large-scale DNA circuits over the microfluidic platform. This flow can help the researchers to evaluate their ideas for future trends in DNA logic design.

6 Simulation Results

This paper introduced an automatic design flow for design and verification of the large DNA circuits based on digital microfluidic biochips. Table 1 illustrates the DNA cluster (DC) output concentration for C17 benchmark with 8 synthesized DCs for “10110” inputs. Each row of this table shows the information of a specified DNA cluster. Column Bitstream shows the synthesized configuration bitstream/Boolean function of the DC. Columns IC1, IC2, IC3 and IC4 show the concentration of Inputs and column OC represents the concentration of output strand for the DC that is resulted after VisulaDSD simulation tool. Finally, column LC show the logical correctness of the DC output (Logic is correct if LC 'Y' (YES)). It is worth to note that high-concentration ($\geq 800nM$) shows logic ‘1’ and low-concentration ($\leq 100nM$) represents logic ‘0’ in this table.

Table 1 shows that the logical output of the DC is correct. Moreover, the output concentration does not degrade with increasing the logic circuit size and noise margin is acceptable for each logic circuit.

Figure 14 shows the details of output concentration for the C17 benchmark that are generated by VisulaDSD. Green and blue curves show the output concentration for “00011” and “10100” inputs, respectively. These graphs prove the logical correctness of the synthesized circuit. Moreover, these curves show the noise margin of the output signals. It is worthwhile to note that DCs are simulated according to DCPG shown in Figure 11. Therefore, at the first step, DC2 and DC3 are simulated. The simulation results of DC3 are sent to DC4, DC5, and DC6 after they are simulated, DC1 is simulated by using DC2 and DC4 outputs and DC0 is simulated using DC5 and DC6 outputs.

Table 2 shows the results of synthesis of some other IBM benchmarks [37] using RTL2DNA flow. In this table, column Complexity and #IO shows the number of gates and IOs, respectively. Column #DC represents the number of DNA clusters in the synthesized circuit and finally, column #Strand shows the required number of DNA strands in the implemented circuit.

As shown in Table 2, the number of required strands is constant for all the circuits in the proposed design flow because the number of strands inside each DNA cluster is independent of other DCs. This property is very critical in feasibility of implementing the real large circuits using DNA logic gates.

Table 3 shows the physical implementation of IBM benchmarks and two full adders with 2 and 6 input bits on microfluidic biochip. The synthesized circuits are implemented on a Programmable Bio-Cell Matrix (PBCM) architecture [38] using SSS toolbox. In Table 3, column Dimension shows the number of rows and columns of the base biochip. Columns #Electrode and #Pin rep-
resent the number of electrodes and number of pins, respectively and finally, column Total time shows the total time of bioassay on the MFBC.

As can be seen in Table 3, physical properties of implementing benchmarks can be evaluated using RTL2DNA flow. Physical properties in addition to logical verification information enable the researchers to evaluate and compare various design choices quantitatively.

7 Conclusion

DNA computing is a fascinating multi-disciplinary technique to utilize the highlighted features of DNA strands for logic design. Considerable methods are addressed to design the DNA-based logic circuits but no practical tool chain is proposed for this purpose. In this paper, an automatic design flow and corresponding tool chain are proposed to facilitate the design and evaluation of multi-stage large-scale DNA logic circuits on the microfluidic biochip platform. We used VTR tool for circuit partitioning then develop a tool chain to map the output of VTR to DNA strands. Placement and routing of DNA circuits on microfluidic platform is done by using the SSS tool. Results show the usability, simplicity and flexibility of the proposed design flow.

An important contribution of the proposed technique is that the number of required strands is constant for all the circuits and does not increase for the large circuits. The proposed tool chain can open a horizon for design, implementation and evaluate the logical DNA circuits on the microfluidic biochip.

References

1. Currin A., Korovin K., Ababi M. et al., "Computing exponentially faster: implementing a non-deterministic universal Turing machine using DNA", In Journal of the Royal Society Interface, Vol.14, No.128, 2017.
2. Adleman L., "Molecular computation of solutions to combinatorial problems", In Science, Vol.266, No.5187, 1994.
3. Sanches C., Soma N., "A general resolution of intractable problems in polynomial time through DNA computing", In Biosystems, Vol.150, No.1, 2016.
4. Zhao K., Wang Z., Lu Y. et al, "A new biologically DNA computational algorithm to solve the k-vertex cover problem", In Journal of Computational and Theoretical Nanoscience, Vol.12, No.3, 2015.
5. Seeman N., "Nanomaterials based on DNA", In Annual review of biochemistry, Vol.79, No.1, 2010.
6. Sakamoto K., Gouzu H., Komiya K. et al, "Molecular computation by DNA hairpin formation", In Science, Vol.288, No.5469, 2000.
7. Livstone M. and Landweber L., "Mathematical considerations in the design of microreactor-based DNA computers", In International Workshop on DNA-Based Computers, 2004.
8. Qian L. and Winfree E., Winfree, "Parallel and scalable computation and spatial dynamics with DNA-based chemical reaction networks on a surface", In International Workshop on DNA-Based Computers, 2014.
9. Fan D., Wang K., Zhu J. et al, "DNA-based visual majority logic gate with one-vote veto function", In Chemical science, Vol.6, No.3, 2015.
10. Qian L. and Winfree E., "A simple DNA gate motif for synthesizing large-scale circuits", In International Workshop on DNA-Based Computers, 2008.
11. Hemphill J. and Deiters A., "DNA computation in mammalian cells: microRNA logic operations" In Journal of the American Chemical Society, Vol.135, No.28, 2013.
12. Wu L., Xiaogang Qu., "Cancer biomarker detection: recent achievements and challenges", In Chemical Society Reviews, Vol.44, No.10, 2015.
13. Qian L. and Winfree E., Bruck J., "Fast online synthesis of generally programmable digital microfluidic biochips", In Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, 2012.
14. Van D., "A programmable molecular computer in microreactors", In International Workshop on DNA-Based Computers, 2004.
15. Dwyer C., "Computer-aided design for DNA self-assembly: process and applications", In IEEE/ACM International Conference on Computer-Aided Design, 2005.
16. Rogers W., Shih W., and Manoharan V., "Using DNA to program the self-assembly of colloidal nanoparticles and microparticles", In Nature Reviews Materials, Vol.1, No.3, 2016.
17. Botti, S., Rufoloni A., Laurenzi S. et al., "DNA self-assembly on graphene surface studied by SERS mapping", In Carbon Journal, Vol.109, No.1, 2016.
18. Amos M., DNA computation, Doctoral dissertation, University of Warwick, 1997.
19. Hua Z., Rouse J.L., Eckhardt A.E. et al, "Multiplexed real-time polymerase chain reaction on a digital microfluidic platform", In Analytical chemistry, Vol.82, No.6, 2010.
20. Chang Y.H., Lee G.B., Huang F.C. et al, "Integrated polymerase chain reaction chips utilizing digital microfluidics", In Biomedical microdevices, Vol.8, No.3, 2006.
21. Cannon B.L., Kellis D.L., Davis P. et al, "Excitonic AND logic gates on DNA brick nanobreadboards", In ACS photonics, Vol.2, No.3, 2015.
22. Christopher B. and Landers J.P., "Electrode materials in microfluidic systems for the processing and separation of DNA: A mini review", In Micromachines, Vol.8, No.3, 2017.
Zohre Beiki received the B.Sc. degree in Computer Engineering from Islamic Azad University of Mashhad and the M.Sc. degrees in computer Engineering from University of Science & Technology, Tehran, Iran in 2008 and 2011, respectively. She is currently an Assistant Professor in Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran and her research interests are DNA computing and embedded system design.

Ali Jahanian received the B.Sc. degree in Computer Engineering from University of Tehran, Tehran, Iran in 1996 and the M.Sc. and Ph.D. degrees in Computer Engineering at Amirkabir University of Technology, Tehran, Iran in 1998 and 2008, respectively. He joined Shahid Beheshti University, Tehran, Iran in 2008. His current research interests are focused on VLSI design automation and DNA Computing.
Figure and table captions

[1] Fig. 1 A simple view of Toehold-mediated strand displacement.

[2] Fig. 2 Internal strands and operations of the AND/OR seesaw logic gate.

[3] Fig. 3 The structure of a digital microuidic biochip.

[4] Fig. 4 A simplified view of the DENA architecture proposed in [22].

[5] Fig. 5 A) Structure of two-level AND/OR DLB in [22] B) Improved DLB which implements a 4-input lookup table.

[6] Fig. 6 Structure and internal strands of amplifying gate [34].

[7] Fig. 7 Overall process of the proposed automatic design flow.

[8] Fig. 8 Design stages from RTL to DNA Netlist for C17.

[9] Fig. 9 The proposed algorithm for DSD model generation from synthesized netlist.

[10] Fig. 10 Generated files for DENA Cluster 0 of IBM C17 benchmark.

[11] Fig. 11 The proposed algorithm for generating the priority graph.

[12] Fig. 12 Simulation steps for each DENA Cluster.

[13] Fig. 13 Generated Architecture and assay file for SSS tool using RTL2DNA flow.

[14] Table 1 Output logic and final output concentration of 8-bit binary adder.

[15] Fig. 14 Concentration level of output signals of C17 Benchmark resulted by VisualDSD.

[16] Table 2 statistical information of attempted benchmarks.

[17] Table 3 Physical properties of attempted benchmarks after realizing on the MFBC platform.
As illustrated in Figure 3, the structure of a digital microfluidic biochip [27] is shown.

Fig. 1 A simple view of Toehold-mediated strand displacement.

Fig. 2 Internal strands and operations of the AND/OR seesaw logic gate [10].
In this paper, the volume of undesirable reactions grows exponentially by increasing the number of gates and complexity of circuits before, the improved DLB contains of 16 number of 5 input AND gates. Therefore, 80 orthogonal DNA strands are required of input lookup table. The following subsections describe the improvements applied to conventional DENA microarchitecture.

Fig. 4 A simplified view of the DENA architecture proposed in [22].

Fig. 5 A) Structure of two-level AND/OR DLB in [22] B) Improved DLB which implements a 4-input lookup table.
Overall, the Berkeley ABC tool [33] is used as technology to synthesize circuit maps. We converted the synthesized gate netlist to the DENA microfluidic platform. We revised the SSS toolbox and visualization to can transform the DLB output to an arbitrary input of the next DLB. This modification improves the logic utilization.

The threshold concentration of the OR gate is between n inputs OFF and 1 input ON, therefore the relative threshold concentration is 16 nM when inputs are ON. In addition, DLB has a 16 \( \frac{1000}{8} \) sub-tree of the first input of the next DLB.

We utilized 8 amplifying gates in this paper. Each amplifying gate received one input (A to D or \( \bar{A} \)).

Physical design:
- The right figure represents the structure and internal strands of the amplifying gate [33].
- The left figure shows the internal strands for an 8-output amplifying gate.

Simulation and verification:
- We used the physical design of the synthesized circuit on the microfluidic platform to generate a placed and routed circuit on the microfluidic platform that realizes the projected circuit on this platform. Finally, a placed and routed circuit on the microfluidic platform is generated that realizes the projected circuit on this platform.

Fig. 6 Structure and internal strands of the amplifying gate [33].

Fig. 7 Overall process of the proposed automatic design flow.
Primary Output implementation of the synthesized circuit that can be simulated using VisualDSD.

It is worth noting that the toolbox should be configured according to circuit primary inputs determined after the first step, DLB.DNA file is created.

At the first step, DLB.DNA file is created. 
Step 2: DLB.DNA file is created. 
Step 3: TV-Pack file is created. 
Step 4: CONV-Pack file is created.

Fig. 8 Design stages from RTL to DNA Netlist for C17.

Fig. 9 The proposed algorithm for DSD model generation from synthesized netlist.
Fig. 10 Generated files for DENA Cluster 0 of IBM C17 benchmark.

Fig. 11 The proposed algorithm for generating the priority graph.
ith the ion (<100nM) represents 0.

Therefore, noise margin is OK for each logic circuit that is a valuable contribution for the proposed method. 

As mentioned before, v.

Step4: DCPG created according to classified node in previous step. Primary nodes have first priority after secondary input) and finally node.

DENA Clusters are classified into three categories: primary nodes have only primary input.

Distributed power allocation to implement the circuit on DMFB.

After that the placement algorithm executes and determines the location of nodes.

The synthesized circuit is implemented on a digital microfluidic biochip.

A prototype microfluidic biochip with 32x32 biochips was used for the test. The test setup included a Droplet Dispenser (DLB), a Mixer-Converter (MIXCONV), a Dispenser (DISPENSE), an Inverter (INV), a Converter (CONV), and two biocompatible microfluidic inverter steps (INV1, INV2) to implement an inverter circuit.

Fig. 12 Simulation steps for each DENA Cluster.

Fig. 13 Generated Architecture and assay file for SSS tool using RTL2DNA flow.
Table 1 Output logic and final output concentration of 8-bit binary adder.

| DC ID | Bitstream (HEX) | DC (nM) | IC1 (nM) | IC2 (nM) | IC3 (nM) | OC (nM) | LC |
|-------|----------------|---------|----------|----------|----------|---------|----|
| 2     | 1000           | 100     | 900      | 100      | 100      | 75      | Y  |
| 3     | 0111 (12, 1, 4, 8) | 100 | 900      | 100      | 100      | 900     | Y  |
| 4     | 1000           | 900     | 75       | 100      | 100      | 76      | Y  |
| 5     | 1000           | 900     | 75       | 100      | 100      | 80      | Y  |
| 6     | 1000           | 900     | 75       | 100      | 100      | 75      | Y  |
| 1     | 1110 (4, 6, 12) | 80      | 75       | 100      | 100      | 75      | Y  |
| 0     | 1110 (4, 6, 12) | 900     | 100      | 100      | 100      | 900     | Y  |

Fig. 14 Concentration level of output signals of C17 Benchmark resulted by VisualDSD.
Table 2  Statistical Information of attempted benchmarks.

| Benchmark | Complexity | #IO | #DC | #Strand |
|-----------|------------|-----|-----|---------|
| C17       | 6          | 7   | 7   | 334     |
| C432      | 160        | 43  | 182 | 334     |
| C1908     | 880        | 58  | 294 | 334     |
| C3540     | 1669       | 72  | 968 | 334     |
| C6288     | 2406       | 64  | 1820| 334     |

Table 3  Physical properties of attempted benchmarks after realizing on the MFBC platform.

| Benchmark | Dimension | #Pin | #Electrode | Total time (ns) |
|-----------|-----------|------|------------|-----------------|
| 2-bit adder | 12 × 21  | 22   | 63         | 21.33           |
| 6-bit adder | 12 × 30  | 52   | 215        | 77.63           |
| C17       | 12 × 25   | 32   | 116        | 66.19           |
| C432      | 12 × 75   | 102  | 457        | 160.24          |
| C1908     | 12 × 128  | 138  | 661        | 893.42          |