DNA-Enabled Integrated Molecular Systems for Computation and Sensing

Craig LaBoda,[†] Heather Duschl,[†] and Chris L. Dwyer*[†,‡]

[†]Department of Electrical and Computer Engineering and [‡]Department of Computer Science, Duke University, Durham, North Carolina 27708, United States

CONSPECTUS: Nucleic acids have become powerful building blocks for creating supramolecular nanostructures with a variety of new and interesting behaviors. The predictable and guided folding of DNA, inspired by nature, allows designs to manipulate molecular-scale processes unlike any other material system. Thus, DNA can be co-opted for engineered and purposeful ends.

This Account details a small portion of what can be engineered using DNA within the context of computer architectures and systems. Over a decade of work at the intersection of DNA nanotechnology and computer system design has shown several key elements and properties of how to harness the massive parallelism created by DNA self-assembly. This work is presented, naturally, from the bottom-up beginning with early work on strand sequence design for deterministic, finite DNA nanostructure synthesis. The key features of DNA nanostructures are explored, including how the use of small DNA motifs assembled in a hierarchical manner enables full-addressability of the final nanostructure, an important property for building dense and complicated systems. A full computer system also requires devices that are compatible with DNA self-assembly and cooperate at a higher level as circuits patterned over many, many replicated units. Described here is some work in this area investigating nanowire and nanoparticle devices, as well as chromophore-based circuits called resonance energy transfer (RET) logic. The former is an example of a new way to bring traditional silicon transistor technology to the nanoscale, which is increasingly problematic with current fabrication methods. RET logic, on the other hand, introduces a framework for optical computing at the molecular level.

This Account also highlights several architectural system studies that demonstrate that even with low-level devices that are inferior to their silicon counterparts and a substrate that harbors abundant defects, self-assembled systems can still outperform conventional systems. Further, the domain, that is, the physical environment, in which such self-assembled computers can operate transcends the usual limitations of silicon machines and opens up new and exciting horizons for their application.

This Account also includes a look at simulation tools developed to streamline the design process at the strand, device, circuit, and architectural levels. These tools are essential for understanding how to best manipulate the devices into systems that explore the fundamentally new computing domains enabled by DNA nanotechnology.
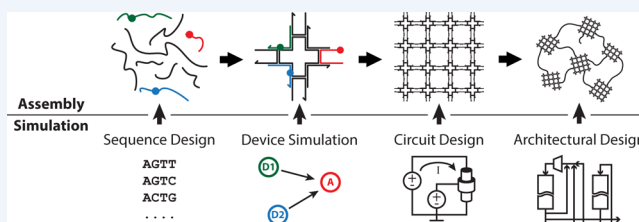
1. INTRODUCTION

This Account reviews work at the intersection between DNA nanotechnology and computer science and engineering. Advances made over the past decade in the programmable assembly of nucleic acids have enabled new domains for computation and challenge our assumptions about how computer systems should take advantage of such technology. In fact, traditional designs for computers, sensors, and memory systems are poorly matched to many important aspects and properties of DNA nanotechnology. Traditional DNA computing uses DNA as the primary computational element, for example, strand displacement[1] or deoxyribozyme-based gates.[2] This work, however, embraces the self-assembly of DNA nanostructures to organize nanoscale devices for computation using the DNA as a scaffold. This Account also describes how these nanoscale devices may be arranged to perform computation independent of DNA strand displacement as a mechanism for actuation[3] and computing.

Ultimately, this Account provides an overview of research in computer engineering focused on how to design, build, and test nanoscale computational systems using chemical self-assembly. This helps bring into focus the differences between conventional silicon-based and nucleic acid technologies and thus highlight new ways to apply the very powerful features of self-assembly to the art of building computers.

The remainder of this Account is organized bottom-up from the physical assembly techniques common in DNA nanotechnology, then into several active components with which we can construct computational systems, and through computer architectures that are uniquely enabled by self-assembly. Each section covers a thin vertical slice through the research community to highlight one possible path from the low-level

physical assembly (chemistry) upward toward useful, sophisticated computer architectures. There are other paths with great promise, and the field has experienced remarkable growth and progress over the past decade. No doubt the coming years will see even greater achievements for the field and bring the promise of DNA nanotechnology into even tighter focus.

2. FULLY ADDRESSABLE HIERARCHICAL ASSEMBLIES

DNA self-assembly offers a bottom-up fabrication mechanism for building integrated circuits at the molecular scale. Computational devices can be chemically conjugated to different strands of DNA that are then self-assembled according to strict Watson–Crick binding rules (i.e., A binds to T; C binds to G). The resulting nanostructure serves as a molecular substrate that organizes devices into desired arrangements. This section describes how DNA nanostructures can be intelligently designed to incorporate computational elements and assembled hierarchically to build systems.

Historically, silicon-based computers have been fabricated using top-down methods. The common feature of a top-down process is a continuous reduction in the characteristic length scale of material structure from the macroscale to the molecular scale.[4] Standard integrated circuit fabrication begins with a silicon substrate upon which various layers of materials are deposited or removed to fabricate computational devices. As device sizes approach the few-atom limit, difficulties in structural and compositional control arise. In contrast, bottom-up fabrication, like DNA self-assembly, builds structures from a set of precursors reacting with a nucleation site, and the structure then grows under thermodynamic control.[4] A set of single-stranded DNA (ssDNA) sequences can be designed such that when mixed together the strands assemble into rigid, complex nanostructures. This mechanism is called "programmable" assembly since the user has control over the formation of the target nanostructure through careful design of the DNA sequences. To build integrated systems using this technology, the designer chemically modifies specific strands before assembly with computational devices through a conjugation reaction. (A variety of devices exist, ranging from nanowires and nanotubes to chromophores, discussed in section 3.) The physical self-assembly process then positions these devices relative to one another to form the desired interactions for the integrated system.

It is well established in computer science and engineering that real, practical computational systems require a large degree of complexity and aperiodicity. To achieve this, complex DNA nanostructures can be hierarchically assembled using building blocks or motifs. By inclusion of short ssDNA overhangs, or sticky-ends, at the edges of a motif, motifs can be programmed to bind to one another and form complicated and dense patterns. For example, the cruciform (Figure 1a), originally designed by Yan et al.[5,6] is composed of nine strands: a core, four shells, and four sticky-ended arms. Sixteen motifs were designed to form a 4 × 4 grid, which can further assemble with other grids to fabricate larger macromolecules as shown in Figure 1b.[7]

Well-designed motifs simplify the building of large DNA structures by reducing the possibility of misassembly. For a system to form properly, motifs that carry computational devices must assemble into a single stable and thermodynamically favorable DNA nanostructure. Thus, sequences should be chosen that maximize the strength of intentional strand interactions and minimize unintentional interactions. One
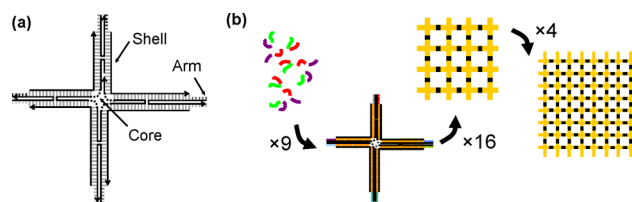
Figure 1. (a) DNA cruciform motif formed by nine unique strands[6] and (b) hierarchical assembly of cruciform motifs to form 4 × 4 grids, which can then by assembled together into even larger arrays.[7] Reproduced with permission from ref 7. Copyright 2007 IOP Publishing.

way to characterize this interaction is to plot the maximal melting temperature, T_m, between each pair of strands over all possible interactions (a computationally expensive prospect for even simple systems), for example, as shown in Figure 2.[6] The
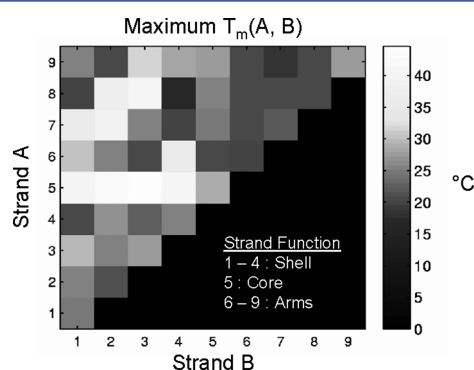

Figure 2. Plot comparing the melting temperatures between the nine strands that form the cruciform motif (shells, core, and arms); strand pairs with the highest melting temperatures are the most favorable interactions and represent the fold-signature for the cruciform.[6]

plots represents a "fold-signature" of the motif, indicating which strand pairs are most thermodynamically favorable, and is compared with an ideal signature based on the intended strand interactions. Additionally, a motif design can be ranked according to thermodynamic stability and specificity metrics.[4] These metrics are useful when choosing among otherwise seemingly equivalent designs; for example, by these metrics random sticky-end sequences often exhibit low specificity, an often-overlooked property that has implications for very complex motif systems.[4]

An ideal nanostructure for assembling a molecular circuit must offer "full molecular addressability" to enable the placement of circuit components at any location. Full addressability on a 4 × 4 DNA grid has been demonstrated, see Figure 3, for example, by spelling out the letters "D", "N",
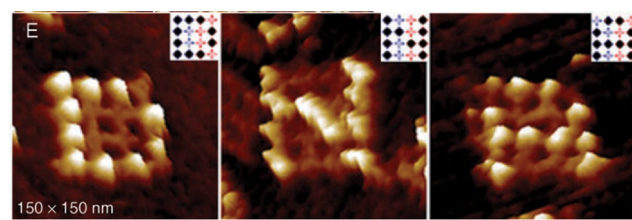
Figure 3. AFM images of DNA grids that are functionalized with biotin–streptavidin to spell out "D", "N", and "A", which demonstrates molecular-scale patterning and full addressability.[8]
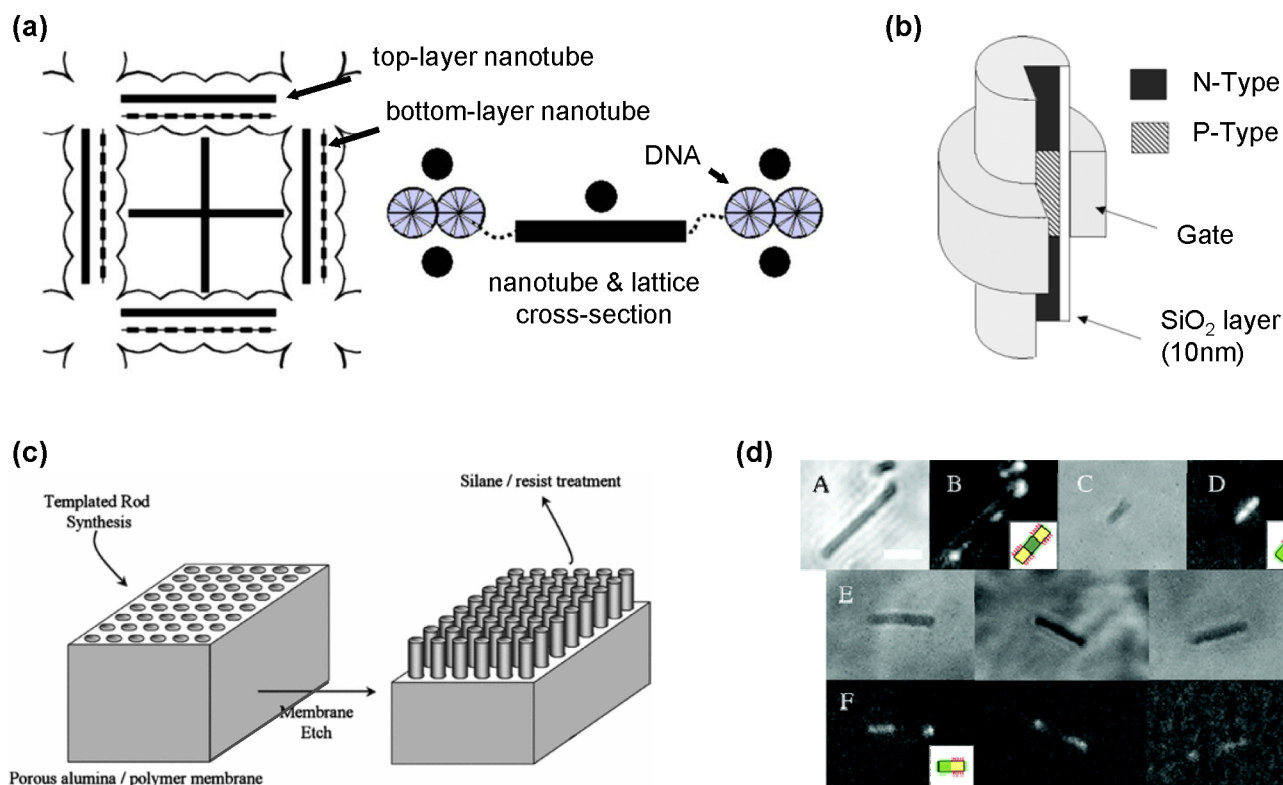
Figure 4. (a) Schematic of a CNT crossbar device within the cavity of a DNA grid, top and side views. Reproduced with permission from ref 9. Copyright 2004 IOP Publishing. (b) Structure of RG-FET and (c) RG-FET synthesis scheme showing selective etching technique. Reproduced with permission from ref 12. Copyright 2008 IEEE. (d) Images of nanorods with fluorescently labeled DNA attached in regions indicated in green on the schematic insets. Reproduced with permission from ref 13. Copyright 2006 American Chemical Society.
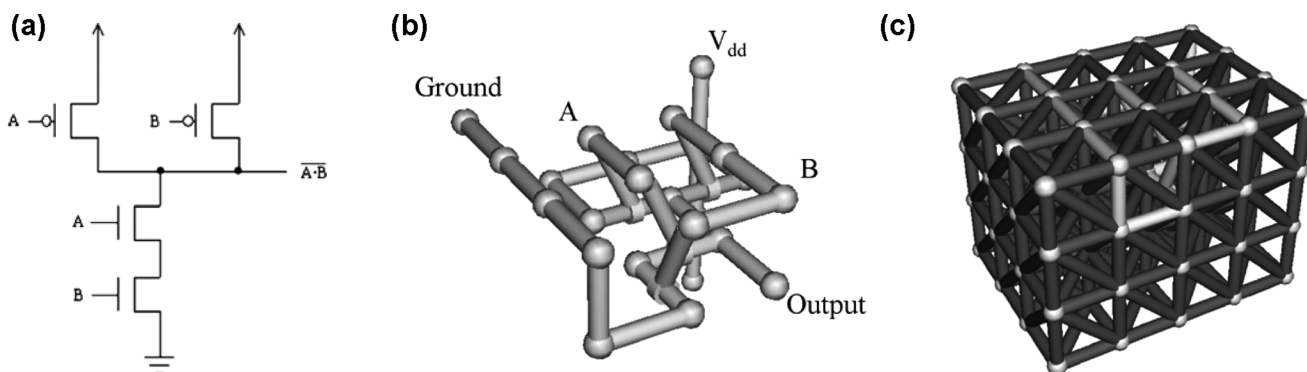

Figure 5. (a) Complementary metal oxide semiconductor (CMOS) NAND gate, (b) conducting portions of RG-FET NAND gate connected together with single stranded DNA, and (c) RG-FET NAND gate with added insulators for structural uniformity and integrity. Reproduced with permission from ref 12. Copyright 2008 IEEE.

and "A" with the protein streptavidin.[8] One method for achieving this is to assign a unique sequence to each strand in a nanostructure. However, this solution limits the size of the DNA structure due to the finite number of orthogonal sequences available.[5] Instead, hierarchical assembly techniques can be employed that retain full addressability of the resulting nanostructures but allow for the reuse of core sequences. In the case of 4 × 4 grids, each tile is assembled in a separate vial before being mixed to form the grid. The core and shell strands, which are hybridized during tile assembly, can be reused for all 16 tiles. This hierarchical scheme reduces the strand-design problem, enabling designers to focus on determining suitable sticky-ends.

3. NANOSCALE COMPUTING ELEMENTS

The full addressability of self-assembled DNA nanostructures can be used to fabricate bottom-up computing systems at the nanoscale. Various circuit elements can be functionalized with ssDNA, which enables their exact placement on a DNA nanostructure. In this way, the DNA substrate acts as a nanoscale equivalent of an electronics breadboard or scaffold. By attaching ssDNA to a nanoparticle and the complementary sequence to the DNA grid, elements can be arranged programmatically. To provide concrete examples of this, we discuss several computational elements functionalized with DNA in this section, including carbon nanotubes,[9] nanowires,[10] proteins,[7] and small light absorbing molecules called chromophores.[11]
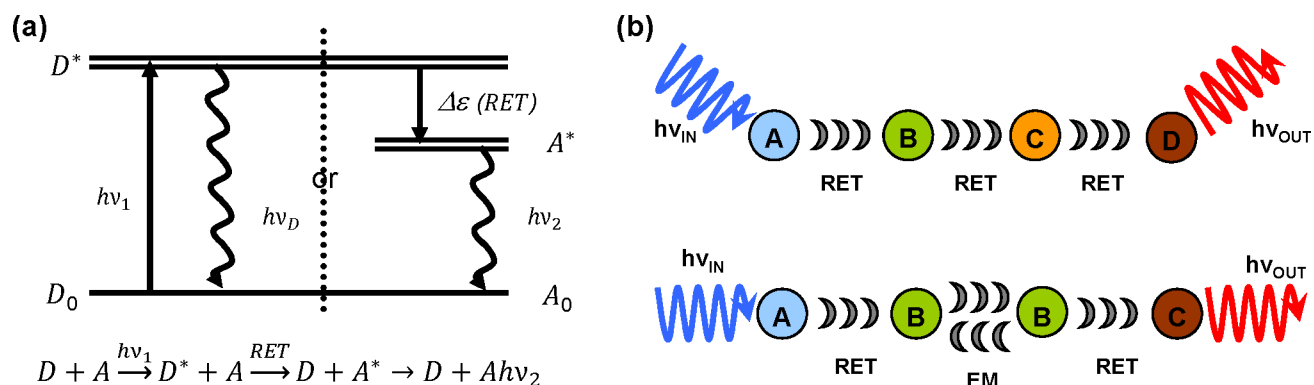
Figure 6. (a) RET energy diagram and equation and (b) RET and EM wires (energy transfers from high to low energy fluorophores, blue to red). Reproduced with permission from ref 11. Copyright 2008 IEEE.
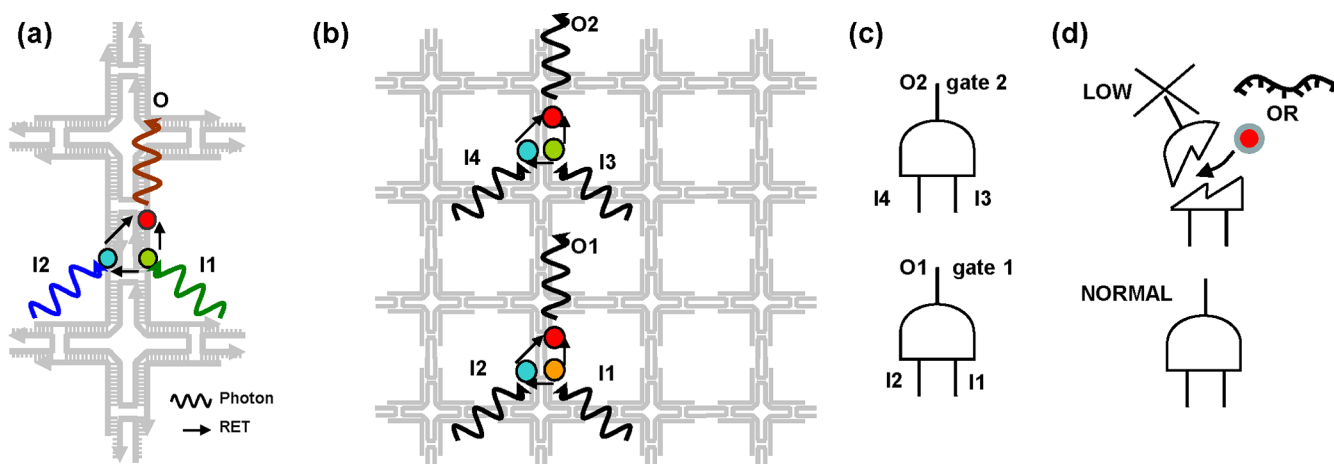
Figure 7. (a) RET AND gate arranged on DNA grid, (b) multiple independently wavelength-addressed AND gates, (c) circuit representation, and (d) AND gates that are broken by small molecules or single stranded nucleic acids (this interrupts RET, changing the output and allowing the gate to act as a sensor).[17]

3.1. Carbon Nanotubes and Ring-Gated Field Effect Transistors

Two useful inorganic elements that lend themselves well to being integrated into DNA nanostructures are carbon nanotubes (CNTs) and nanowires. The cavities in the DNA grid provide a convenient site to make crossbar devices where metal and semiconducting nanotubes or nanowires meet to form junctions (Figure 4a).[9] When appropriately doped, a nanowire can form a ring-gated field effect transistor (RG-FET) as shown in Figure 4b.[12] This is a nanowire with regions of n-type and p-type semiconductor, a silicon dioxide layer, and a metallic gate ring. Single-stranded DNA can be attached to the nanowire at any position: on the gate electrode, at the ends of the wire, or along the wire length. RG-FETs can be synthesized by either a selective etching method shown in Figure 4c[12] or a sequential deposition method.[13] Results of selective functionalization by the sequential deposition method are shown in Figure 4d[13] where DNA functionalized regions are illuminated by a DNA-binding fluorescent dye. Metalization of the DNA[14] after the integration of the crossbar or RG-FET components completes the circuit.

With RG-FETs functionalized at the ends with single-stranded DNA and metalized double-stranded DNA rods, arbitrary logic could in principle be translated from a circuit diagram to a cubic cell 3D self-assembled structure. This represents an extreme in the design space where the DNA is used simply to connect the ends of components. Figure 5 shows this process for a logical NAND gate.[12] By tailoring the complementarity of the ssDNA regions on the ends of the nanorods, we could assemble the structure into the scaffold shown in Figure 5c. Because of the complexity of the structure and the limited number of unique DNA binding regions, hierarchical assembly will likely be needed. These cubic cell structures, while theoretically feasible, are very large by molecular-mass standards. Thus, their construction may suffer tremendous yield loss and may not be possible in practice.

3.2. Resonance Energy Transfer Devices

DNA nanotechnology also enables self-assembled nanoscale optical circuitry. Chromophores, or molecules that absorb light at a certain wavelength, can be arranged on DNA nanostructures to create nanophotonic networks that undergo nonradiative, near-field energy transfer.

This process is known as resonance energy transfer (RET) and occurs when the absorption spectrum of one chromophore (i.e., an acceptor) overlaps with the emission spectrum of another chromophore (i.e., a donor) and they are within nanometers of each other. The efficiency of RET scales as $1/r^6$, where r is the separation between the donor and acceptor.[15] Arranging multiple donor−acceptor pairs into a specific pattern creates a RET network that can perform logic or analogue functions. Photons supplied by an external source act as inputs to these networks, and output fluorescence is recorded with a
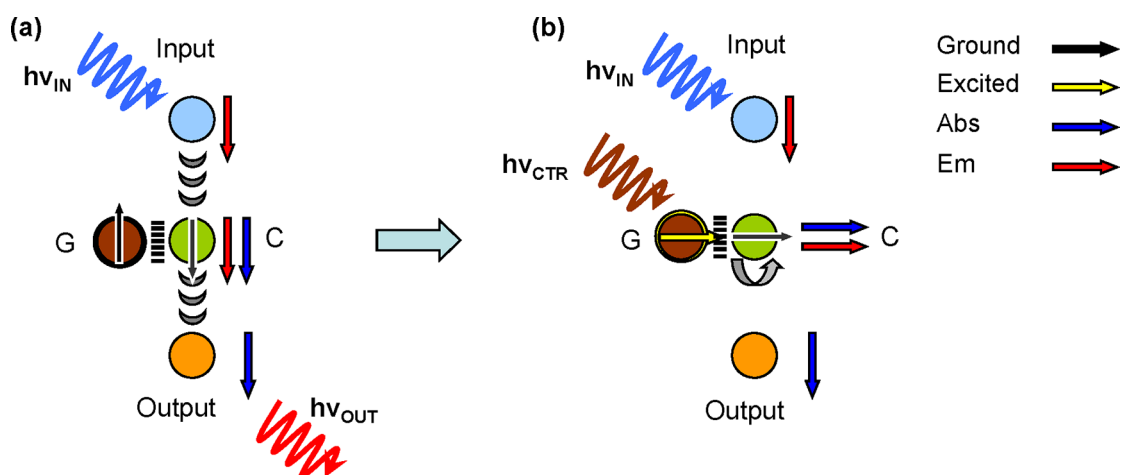
Figure 8. RET pass gate in (a) pass mode and (b) high-impedance mode. Adapted with permission from ref 11. Copyright 2008 IEEE.
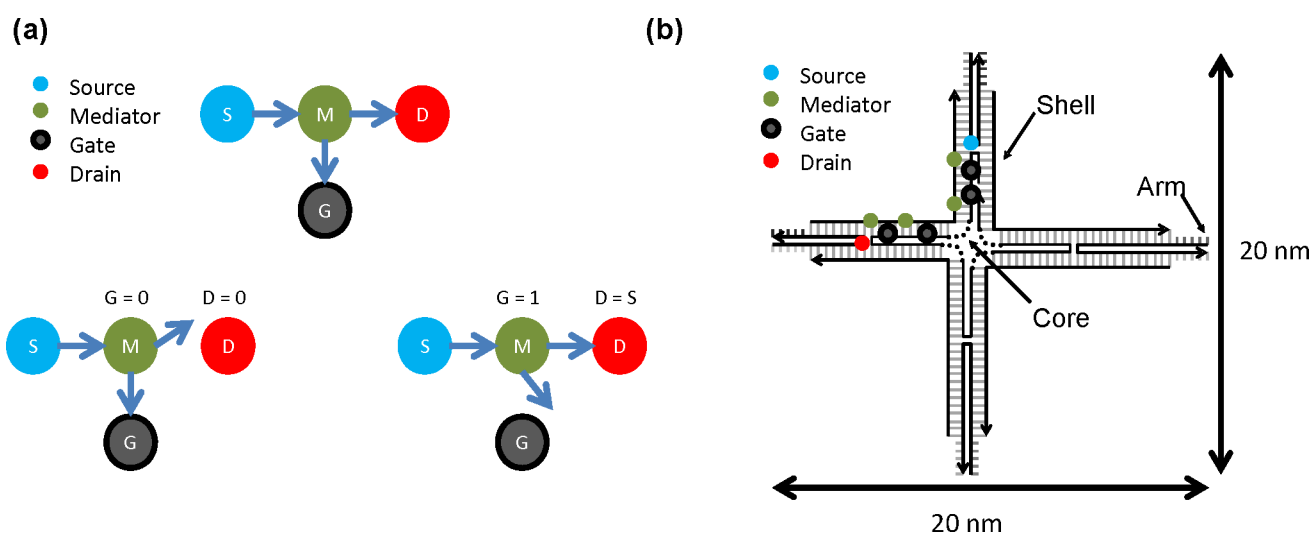

Figure 9. (a) Schematic of C-DEV behavior. In the closed state, energy flows from source to gate. When the gate is externally excited (G = 1), the gate opens, and energy flows from source to drain. (b) Location of C-DEV elements on DNA tile.[20]

photomultiplier or single photon avalanche detector. An energy transition diagram illustrating RET is shown in Figure 6a.[11] First, a photon with energy $h\nu_1$ is absorbed by the donor, D, and promotes it to an excited state, D*. D* then either emits a photon with energy $h\nu_D$ or transfers its energy to the acceptor A via RET, thereby promoting A to A*. The A* species can then decay nonradiatively or emit a photon of energy $h\nu_2$. When chromophores are too far away, they cannot efficiently participate in RET and instead act as independent systems.

The simplest RET network is an energy cascade that acts similarly to a lossy wire moving energy within a circuit. The total efficiency of an n chromophore wire scales as $\varphi_T{}^n$, where φ_T is the transfer efficiency from donor to acceptor.[11] A more efficient wire uses energy migration (EM) where the excited state energy of a donor can readily diffuse within a group of closely packed identical chromophores. When tightly coupled, the group of molecules has the same de-excitation probability as any one donor molecule.[11] Further, by placing an EM wire within a RET cascade, the energy migration will be biased toward the final acceptor due to state occupancy, which provides a drift-diffusion term. Both types of wires are schematically illustrated in Figure 6b.[11]

When two donors with distinct excitation wavelengths are sufficiently close to supply energy to the same acceptor, a logical OR gate is constructed.[16] Exciting either donor results in appreciable output emission from the acceptor. By applying a threshold level to the output and moving the donors away from the acceptor, one can implement an AND gate (Figure 7a−c).[17] With different combinations of input chromophores, multiple gates can be placed on the same nanostructure. The distance dependence of RET enables these logic gates to also act as sensors. By placement of a receptor for an analyte of interest between the donor−acceptor pairs, the binding event with as little as 8 fmol of sample modulates the output (Figure 7d).[17]

Transfer efficiency between a donor and an acceptor relies not only on distance but also on the alignment of dipole moments between chromophores. Each chromophore has four dipoles: permanent ground- and excited-state dipoles, and transient absorption and emission dipoles. By forcing dipoles into or out of alignment with one another, a primitive pass gate[11] can be constructed. Electrostatic interaction between chromophores, which has been demonstrated using alignment of permanent dipoles at nanometer range,[18] in theory allows the transient excited state dipole of one chromophore to realign a neighboring chromophore's permanent ground state dipole. If

chromophores can be found such that the excitation and emission dipoles are parallel or perpendicular to the permanent dipoles, then the relative orientation of chromophores (and thus RET efficiency through the gate) can be controlled by external inputs.[11] The pass gate in Figure 8 uses this concept to modulate the alignment of a channel chromophore (C) with respect to an input and output chromophore. Without an external gate input energy flows from input to output, but when the gate is excited it misaligns the dipole of the channel and prevents RET. These pass gates are still theoretical elements; however, they could enable larger systems of circuits than with RET logic alone.

An advantage of RET logic is that because inputs and outputs are externally supplied, multiple wavelengths of light can be input and read simultaneously. One such multiple-wavelength design uses polychromatic address multiplexing (PAM) to increase optical storage density beyond traditional wavelength-induced limits.[19] This scheme implements a read-only memory storage cell, referred to as a photoerasable PAM element (PEPE), and in its simplest form uses a donor−acceptor pair (or an excitation port−read port pair) where the acceptor is conjugated to DNA using a UV-cleavable linker. When UV light is applied to the cell, the acceptor is released from the DNA, which disables RET. Address multiplexing is achieved by including a set of acceptors that take energy from the excitation port and reduce RET to the read port. When these additional acceptors are excited by an external source, they can no longer take energy from the excitation port (an effect known as state saturation), and RET can occur, providing the cell's output value. By use of unique combinations of photon energies, multiple individual PEPEs can be selected (i.e., addressed), enhancing optical storage media densities by several thousand-fold over DVD/BluRay.

The saturation effect used in PAM also enables logic elements with nonlinear gain. One such structure is the diffusive exciton valve (DEV).[20] This element has two operational modes: normally closed (C-DEV) and normally open (O-DEV). A preliminary C-DEV has been experimentally implemented using four types of chromophores: source, mediator, drain, and gate (Figure 9). As shown in the schematic of Figure 9a,[20] excitons flow from source to mediator and then from mediator to gate or drain. In the default C-DEV state, excitons flow from source to gate with a low signal at the drain chromophore. When the gate chromophore(s) are saturated by external sources, excitons cannot be captured by the gate and instead flow through to the drain (output). This opens the C-DEV, which can have as high as 10^8 greater exciton flow than when it is closed.

The large set of design constraints and complexity inherent in RET networks necessitates the development of an automated design framework. We have created a design flow for RET device networks that begins with an energy flow diagram, desired logic functionality, available binding sites, and available chromophores and produces candidate RET networks within those constraints.[21] Heuristic rules reduce the number of candidate designs without requiring any simulation, and the pruned space is then simulated using a physical model, for example, an algorithm called "Karoon"[22] or a hybrid Monte Carlo/Markov chain model called "SCIMM".[23] With this design flow, it becomes possible to create the complex networks necessary for developing the self-assembled architectures described in the next section.

## 4. COMPUTER ARCHITECTURES AND SYSTEMS

Conventional computer architectures place such stringent tolerance requirements on the device and circuit processes that they are impractical to adopt for self-assembly. However, if we approach the problem without such constraints, we can find alternative designs that take advantage of the many unique and powerful properties of self-assembly. The device elements described in section 3, together with the fully addressable, hierarchical DNA assemblies of section 2, enable the development of many self-assembled computer architectures and systems. These DNA-based architectures offer a variety of advantages over top-down fabricated silicon systems. By taking advantage of the self-assembly process, these systems can sample all instances of an intractable problem space in parallel. Although these systems currently only offer simple computation, the sheer number of assembled devices brings the total number of operations to a level that is computationally relevant. Additionally, their small size and use of organic materials make them better suited for some environments currently inaccessible with silicon technology.

This section begins by describing the architectural challenges introduced by bottom-up assembly. Afterward, a handful of possible architectures are introduced that were chosen to clearly demonstrate the advantages of self-assembly. An execution model, instruction set, and memory model have been developed for each architecture but only the defining characteristics of these systems will be mentioned in this section. All of these architectures are device agnostic unless otherwise specified. Interested readers are referred to the references for a more detailed understanding.

4.1. Architectural Challenges

The use of bottom-up fabrication for building nanoscale architectures presents a new set of challenges for circuit designers. The diminishing yield of large-scale self-assembly (ca. 80% are fully formed, good structures) limits the maximum size and complexity of any single structure. Without the ability to assemble one large system, designers must first fabricate small circuits called nodes, for example, composed of DNA grids patterned with predefined CNT crossbars, RG-FETs, or chromophores. These nodes are then assembled at a larger length-scale into processing elements (PEs) that can execute instructions. The simplest, most feasible to build self-assembled architectures lack large-scale interconnects, and nodes must communicate with only their nearest neighbors (Figure 10). These limited communication pathways are further complicated by the randomness introduced by the self-assembly process. Simple strategies for assembling large systems cannot guarantee the placement, orientation, or connectivity between nodes. Circuit designers can cope with this randomness through proper node and system level design.

Bottom-up fabrication also results in high defect rates due to the probabilistic nature of self-assembly. A wide variety of structural defects at the strand, tile, and grid level have been identified and characterized by AFM.[24] When these errors occur at the node level, they can render a PE incapacitated, semifunctional, or even detrimental to other nodes. At the architectural level, internode connections may be omitted or broken. In many general purpose architectures, a postassembly configuration phase imposes coordination between nodes in order to overcome both node defects and system-level randomness and yield a reasonable abstract machine interface for application programmers.
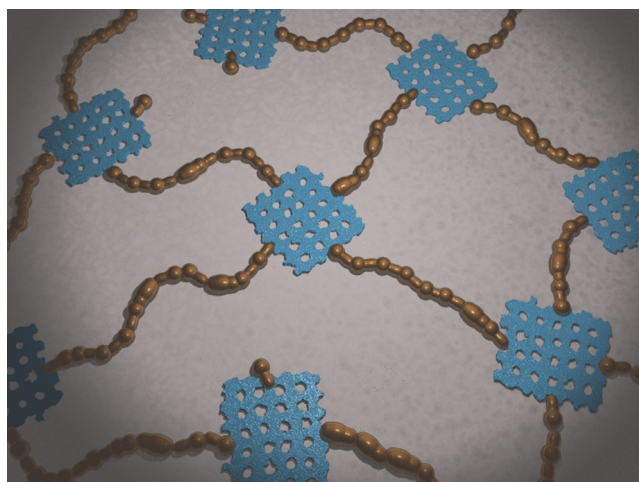
Figure 10. Schematic rendering of a self-assembled DNA network consisting of nodes connected to nearest neighbors. Reproduced with permission from ref 4. Copyright 2007 Artech House, Inc.

4.2. Temporal Aspects

Bottom-up fabrication also presents opportunities for innovation in architectural design. One such opportunity is the ability for self-assembled architectures to compute solutions *during* their fabrication.[25] Conventional computation is typically performed either pre- or postfabrication. In postfabrication computation, processors are first manufactured and then used to solve a variety of problems. General purpose processors such as the Intel core i7 fall into this category. In prefabrication, calculations are performed before the computer is manufactured. The inclusion of a ROM lookup table is one example of prefabrication computation. This specialized hardware stores solutions to common problems so processors do not need to recompute the solution each time the problem is encountered. While both pre- and postfabrication computation are available to self-assembled computers, at-fabrication computation may also be incorporated into the architectural scheme. By cleverly encoding combinatorial problems into the DNA sequences themselves, similar to encodings used in traditional DNA computing,[26] the interactions of the self-assembly process can sample all possible solutions. To find whether a suitable solution exists, the set of assembled structures must be inspected for solutions that meet the problem statement's criteria. The mechanism by which the structures are queried depends on the device technology; however, the concept is general.

The following sections provide specific examples of DNA-self-assembled computer architectures that address the challenges and temporal aspects of bottom-up computing described above.

4.3. Oracle

Oracles are self-assembled architectures that blend at-fabrication and postfabrication computation. Abstractly, an oracle contains a large number of question and answer pairs. If the question posed is contained in any of the oracle's question/answer pairs, a response is generated much like a content addressable memory (CAM).[27] To design an oracle, the question/answer pair space is first encoded into a set of DNA tiles. Figure 11 schematically illustrates the DNA tiles necessary for creating a two-bit Oracle adder.[4] Each tile represents a line from the adder's truth table. Question/answer pairs in this case
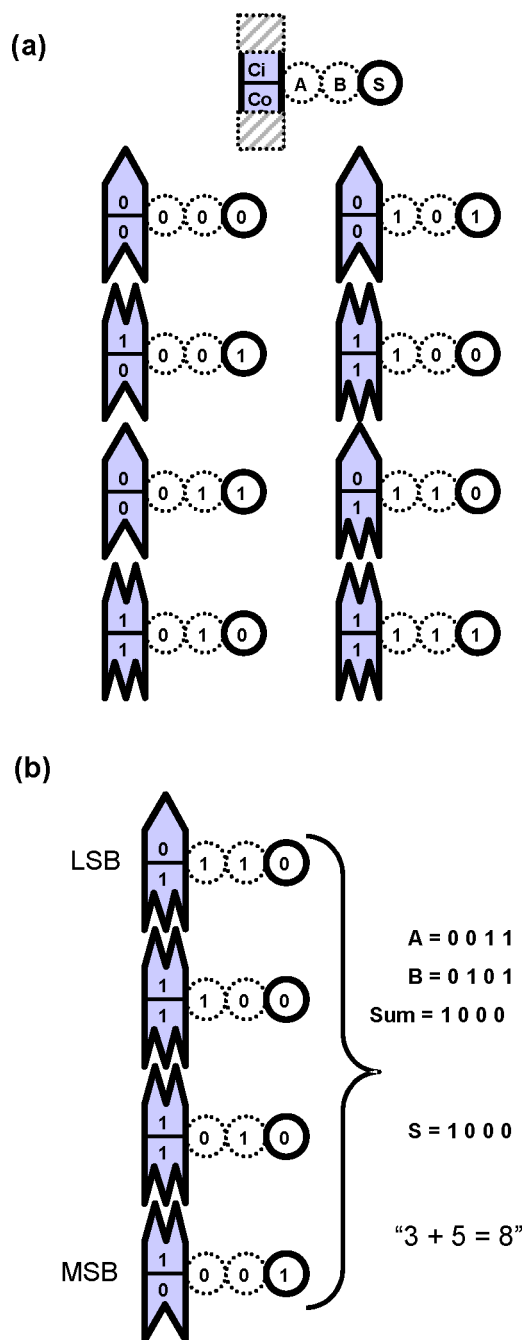
Figure 11. (a) Oracle tile representation of two-bit addition truth table and (b) tile representation of the question "3 + 5 = ?" with the corresponding answer "8". Reproduced with permission from ref 4. Copyright 2008 Artech House, Inc.

are two-bit additions, that is "What is 3 + 5?", and their sums, respectively. During fabrication, the Oracle's tiles assemble according to their binding rules and physically compute every possible solution to the encoded problem, similar to Adleman-style DNA computing. In the adder example of Figure 11, DNA tiles connect together based on their sticky-end configurations represented by the carry-in and carry-out edge shapes, and all possible two-bit additions are assembled. The key difference between Oracles and Adleman-style DNA computing is that each DNA tile carries a set of computational devices, such as the CNTs or RG-FETs described in section 3. Sets of tiles that assemble into a correct solution complete a predefined circuit

that supplies the user with the answer postfabrication. This feature improves the speeds of typical DNA computing systems by offering real-time readout. The Oracle's design has been generalized for use in more computationally intractable scenarios, such as the NP-complete Hamiltonian Path problem.[4]

4.4. Self-Organizing SIMD Architecture

The Oracle's at-fabrication computation enables it to solve problems with vast input spaces in short periods of time; however, its architecture limits it to a small class of problems. The self-organizing SIMD architecture (SOSA) is a general purpose architecture that overcomes the randomness and high defect rates associated with bottom-up fabrication and offers far more flexibility at runtime through postfabrication computation.[28] SOSA is composed of a large array of randomly interconnected nodes that each contain a 1 bit arithmetic logic unit (ALU), a 1 bit data buffer, and a 32 bit register file. Nodes communicate asynchronously, sending single bits across three virtual channels, and configure themselves at power-up into a logical tree structure.[28] Since individual nodes are too simple to perform useful computation, nodes are grouped together to form PEs. A specialized anchor node broadcasts a full instruction (containing multiple microinstructions) to all nodes using the tree established during configuration. Data can then be relayed to an external controller using shift instructions that move information between adjacent PEs. Overall, the asynchronous execution of nodes allows SOSA to achieve high performance. SOSA was evaluated using a custom, event-driven simulator and compared with current processors at the time (Intel Pentium 4 and an ideal superscalar). Results indicated that SOSA could achieve significantly higher throughput using the same computational substrate area while tolerating a high fraction of defective nodes (up to 30%). This work demonstrated that even though the self-assembled technology provided faulty devices, a modern-style, general purpose computer architecture is still possible.

4.5. Nanoscale Sensor Processors

The self-assembled architectures discussed thus far have all been device agnostic and could take advantage of any of the computing components mentioned in section 3. The nanoscale sensor processor (nSP) specifically takes advantage of the chromophore based pass gates to blend computation with sensing.[29] The nSP design was developed for sensing and computing in biological domains unfit for typical silicon processors. Example applications include counting analytes within a cell or more generally monitoring nanoscale biological processes. These applications and domains place strict area requirements on nSPs. To meet this requirement, nSPs employ a simple, accumulator based architecture with a small "unified instruction, data, and sensor memory space" of 256 four-bit words.[29] To conserve space, instructions are variable size and can use instruction-fused sensing (IFS) to directly sense the environment. IFS opcodes are modified by a sensing event in which an analyte binds to a receptor site on the DNA grid. This binding event interrupts RET between chromophores used to store the bits of an instruction, dynamically changing, for example, from a jump to a NOP. By integrating sensing events into the program encoding, IFS can reduce the memory footprint of multisensor systems by as much as 58% as demonstrated by a cycle accurate simulation on a suite of benchmark programs related to biologically relevant sensing tasks.[29]

5. CONCLUDING REMARKS

This Account has presented an overview of a wide range of new devices, circuits, and architectures, enabled by DNA self-assembly. Using bottom-up fabrication as a platform for organizing computational devices, the assembly of RG-FET networks and other nanoelectronics brings traditional computing to the nanoscale with an unprecedented computational density. Similarly, DNA nanostructures provide the molecular breadboard necessary for integrated RET networks that produce a practical, low-cost solution for computing in otherwise inaccessible domains including aqueous and biological environments. At the architectural level, this Account has explored systems that solve many underlying challenges associated with self-assembly, including lack of large-scale control, inherent randomness, and high defect rates of bottom-up fabrication. Ultimately, the topics covered in this Account serve as encouraging evidence that DNA self-assembly can facilitate the production of functional, integrated molecular scale systems and leaves room for innovation in new computational domains.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: c.dwyer@duke.edu.

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. C.L. and H.D. contributed equally.

Notes

The authors declare no competing financial interest.

Biographies

Craig LaBoda is a doctoral student in electrical engineering at Duke University. He has a B.S. in electrical engineering from Boston University and an M.S. in electrical engineering from Duke University. His research interests include DNA self-assembled systems, upconverting nanoparticles, and molecular computing.

Heather Duschl received a B.S. in computer engineering from the University of Pittsburgh and an M.S. in electrical engineering from Duke University. Her research interests have focused on nanotechnology and embedded systems solutions in biology and medicine.

Chris Dwyer is currently an associate professor of electrical and computer engineering and of computer science at Duke University. His research interests include self-assembly, DNA, and the development of device, circuit, and systems technologies for molecular-scale fabrication processes. He has a B.S. in computer engineering from the Pennsylvania State University, University Park, and an M.S. and a Ph.D. in computer science from the University of North Carolina at Chapel Hill.

■ ACKNOWLEDGMENTS

■ REFERENCES

(1) Seelig, G.; Soloveichik, D.; Zhang, D. Y.; Winfree, E. Enzyme-Free Nucleic Acid Logic Circuits. *Science* **2006**, *314*, 1585−1588.

(2) Stojanovic, M. N.; Stefanovic, D. A Deoxyribozyme-Based Molecular Automaton. *Nat. Biotechnol.* **2003**, *21*, 1069−1074.

(3) Graugnard, E.; Kellis, D. L.; Bui, H.; Barnes, S.; Kuang, W.; Lee, J.; Hughes, W. L.; Knowlton, W. B.; Yurke, B. DNA-Controlled Excitonic Switches. *Nano Lett.* **2012**, *12*, 2117−2122.

(4) Dwyer, C.; Lebeck, A. *Introduction to DNA Self-Assembled Computer Design*; Artech House: Norwood, MA, 2007.

(5) (a) Yan, H.; Park, S. H.; Finkelstein, G.; Reif, J. H.; LaBean, T. H. DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires. *Science* **2003**, *301*, 1882−1884. (b) Wei, B.; Dai, M.; Yin, P. Complex Shapes Self-Assembled from Single-Stranded DNA Tiles. *Nature* **2012**, *485* (7400), 623−626.

(6) Dwyer, C.; Park, S. H.; LaBean, T.; Lebeck, A. The Design and Fabrication of a Fully Addressable 8-tile DNA Lattice. *Proc. Found. Nanosci.* **2005**, 187−191.

(7) Pistol, C.; Dwyer, C. Scalable, Low-Cost, Hierarchical Assembly of Programmable DNA Nanostructures. *Nanotechnology* **2007**, *18*, No. 125305.

(8) Park, S. H.; Pistol, C.; Ahn, S. J.; Reif, J. H.; Lebeck, A. R.; Dwyer, C.; LaBean, T. H. Finite-Size, Fully Addressable DNA Tile Lattices Formed by Hierarchical Assembly Procedures. *Angew. Chem.* **2006**, *45*, 735−739.

(9) Dwyer, C.; Johri, V.; Cheung, M.; Patwardhan, J.; Lebeck, A.; Sorin, D. Design Tools for a DNA-Guided Self-Assembling Carbon Nanotube Technology. *Nanotechnology* **2004**, *15*, 1240−1245.

(10) Dwyer, C. Self-Assembled Computer Architecture: Design and Fabrication Theory. Ph.D. Thesis, University of North Carolina at Chapel Hill, May 2003.

(11) Pistol, C.; Dwyer, C.; Lebeck, A. R. Nanoscale Optical Computing Using Resonance Energy Transfer Logic. *IEEE Micro* **2008**, *28*, 7−18.

(12) Dwyer, C.; Vicci, L.; Poulton, J.; Erie, D.; Superfine, R.; Washburn, S.; Taylor, R. M. The design of DNA self-assembled computing circuitry. *IEEE Trans. Very Large Scale Integration* **2004**, *12*, 1214−1220.

(13) Skinner, K.; Dwyer, C.; Washburn, S. Selective Functionalization of Arbitrary Nanowires. *Nano Lett.* **2006**, *6*, 2758−2762.

(14) Richter, J.; Mertig, M.; Pompe, W.; Mönch, I.; Schackert, H. K. Construction of Highly Conductive Nanowires on a DNA Template. *Appl. Phys. Lett.* **2001**, *78*, 536−538.

(15) Valeur, B. *Molecular Fluorescence: Principles and Applications*; Wiley-VCH: Weinheim, Germany, 2001; pp 113−123.

(16) Dwyer, C.; Lebeck, A. R.; Pistol, C. Energy Transfer Logic on DNA Nanostructures: Enabling Molecular-Scale Amorphous Computing, Presented at the 4th Workshop on Non-Silicon Computing, 2007

(17) Pistol, C.; Mao, V.; Thusu, V.; Lebeck, A. R.; Dwyer, C. Encoded Multichromophore Response for Simultaneous Label-Free Detection. *Small* **2010**, *6*, 843−850.

(18) Harper, A. W.; Sun, S.; Dalton, L. R.; Garner, S.M.; Chen, A.; Kalluri, S.; Steier, W. H.; Robinson, B. H. Translating Microscopic Optical Nonlinearity into Macroscopic Optical Nonlinearity: The Role of Chromophore- Chromophore Electrostatic Interactions. *J. Opt. Soc. Am. B* **1998**, *15* (1), 329−337.

(19) Mottaghi, M.; Dwyer, C. Thousand-Fold Increase in Optical Storage Density by Polychromatic Address Multiplexing on Self-Assembled DNA Nanostructures. *Adv. Mater.* **2013**, *25*, 3593−3598.

(20) Thusu, V. Self-Assembled Resonance Energy Transfer Devices. Ph.D. Thesis, Duke University, April 2013.

(21) Mottaghi, M.; Rallapalli, A.; Dwyer, C. RetLab: A fast design-automation framework for arbitrary RET networks. *Proc. DATE* **2014**, 1−6.

(22) Mottaghi, M.; Dwyer, C. Karoon: A fast and precise RET-simulation algorithm for multi-donor multi-acceptor networks. *J. Biophys.* **2014**, in press.

(23) Dwyer, C.; Rallapalli, A.; Mottaghi, M.; Wang, S. *Nanophotonic Information Physics: Nanointelligence and Nanophotonic Computing*; Naruse, M., Ed.; Springer: New York, 2014; pp 41−65.

(24) Mao, V.; Thusu, V.; Dwyer, C.; Chakrabarty, K. Connecting fabrication defects to fault models and SPICE simulations for DNA self-assembled nanoelectronics. *IET Comput. Digit. Tech.* **2009**, *3*, 553−569.

(25) Dwyer, C.; Lebeck, A. R.; Sorin, D. J. Self-Assembled Architectures and the Temporal Aspects of Computing. *Computer* **2005**, *38*, 56−64.

(26) Adleman, L. M. Molecular Computation of Solutions to Combinatorial Problems. *Science* **1994**, *266*, 1021−1024.

(27) Dwyer, C.; Poulton, J.; Taylor, R.; Vicci, L. DNA Self-Assembled Parallel Computer Architectures. *Nanotechnology* **2004**, *15*, 1688−1694.

(28) Patwardhan, J. P.; Johri, V.; Dwyer, C.; Lebeck, A. R. A Defect Tolerant Self-organizing Nanoscale SIMD Architecture. *Proc. 12th Int. Conf. Arch. Supp. Prog. Lang. Oper. Syst. (ASPLOS-XII)* **2006**, 241−251.

(29) Pistol, C.; Chongchitmate, W.; Dwyer, C.; Lebeck, A. R. Architectural Implications of Nanoscale-Integrated Sensing and Computing. *IEEE Micro* **2010**, *30*, 110−120.