Combinational Logic Design Using Six-Terminal NEM Relays

Daesung Lee, W. Scott Lee, *Student Member, IEEE,* Chen Chen, Farzan Fallah, J Provine, *Member, IEEE,* Soogine Chong, *Student Member, IEEE,* John Watkins, *Member, IEEE,* Roger T. Howe, *Fellow, IEEE,* H.-S. Philip Wong, *Fellow, IEEE,* and Subhasish Mitra, *Fellow, IEEE*

***Abstract*—This paper presents techniques for designing nanoelectromechanical relay-based logic circuits using six-terminal relays that behave as universal logic gates. With proper biasing, a compact 2-to-1 multiplexer can be implemented using a single six-terminal relay. Arbitrary combinational logic functions can then be implemented using well-known binary decision diagram (BDD) techniques. Compared to a CMOS-style implementation using four-terminal relays, the BDD-based implementation can result in lower area without major impact on performance metrics such as delay, and energy (when the relays are scaled to small dimensions). Although it is possible to implement any combinational circuit with a single mechanical delay, the relay count can be significantly reduced for complex logic functions by allowing multiple mechanical delays.**

***Index Terms*—Binary decision diagram, logic synthesis, nanoelectromechanical (NEM) relays, nanotechnology.**

I. Introduction

NANOELECTROMECHANICAL (NEM) relays exhibit two unique characteristics, not available in traditional MOSFETs, for energy-efficient digital system design: zero leakage current and near infinite subthreshold slope [1]–[4]. These characteristics allow zero off-state energy dissipation and aggressive supply voltage scaling for ultralow dynamic energy consumption. Scaling of NEM relays is critical for reducing the actuation voltage, switching delay, switching energy, and device footprint [2].

Various NEM relay implementations are possible [1], [4], [5]. The simplest electrostatically actuated NEM relay consists of three terminals—a source, a gate, and a drain, as shown in Fig. 1(a). By adding a second contact region and an electrically isolated conductive channel on the beam, a fourth terminal (the body) can be added to the relay, as shown in Fig. 1(b). The switching time of a relay is a sum of the mechanical delay and the electrical delay. The mechanical delay of a NEM relay, the time required to mechanically switch a relay from the off- (i.e., source and drain disconnected) to the on-state (i.e., source and drain connected) can range from nanoseconds for scaled relays to tens of microseconds. In contrast, the electrical delay (the time required for capacitance charging and discharging) of a single relay can range from a few picoseconds to hundreds of picoseconds [1], [4], [5]. Therefore, to minimize the overall switching time for a given logic function, the maximum number of mechanical delays on any circuit path must be minimized [4], [5]. For four-terminal relays, the state of a relay is independent of the source voltage but is uniquely determined by the voltage difference between the gate and body [6]. Hence, a logic function can be implemented as a single complex gate, where all relays move simultaneously resulting in a single mechanical delay implementation of the function [4], [5]. Note that, this feature is not available in three-terminal relays because the state of this type of relay is determined by the voltage difference between the source and gate.

A six-terminal relay can be formed from a four-terminal relay by adding a second source and body to the other side of the beam, as shown in Fig. 1(c). This type of relay can be actuated bidirectionally and the electrical connection between each source and the drain can be established by sweeping the gate voltage uniquely while maintaining proper bias voltages on the two bodies.

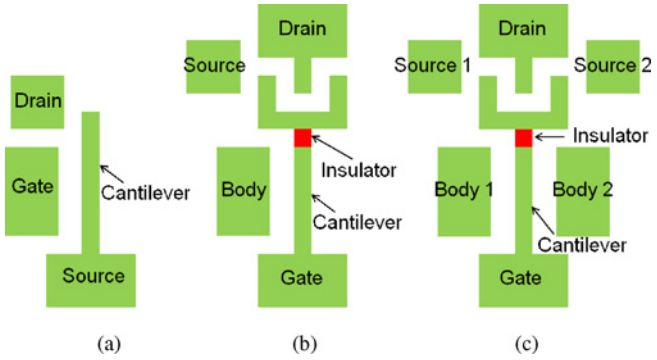


Fig. 1. Schematic of a top view of (a) a three-terminal relay, (b) a four-terminal relay, and (c) a six-terminal relay. The movable cantilever beams can be actuated electrostatically. For the four-terminal and six-terminal relays, an insulating region (shown in red) isolates the contacts from the base of the cantilever.

Manuscript received February 15, 2012; revised June 14, 2012 and September 10, 2012; accepted October 16, 2012. Date of current version April 17, 2013. This work was supported in part by the DARPA/MTO Program Nano Electro Mechanical Computers, under Contract NBCH 1090002 (Program Manager Dr. A. I. Akinwande), and by the National Science Foundation Center for Energy Efficient Electronics Science. This paper was recommended by Associate Editor Y. Chen.

The authors are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: daesung@stanford.edu; wslee@stanford.edu; cchen86@stanford.edu; farzan@stanford.edu; jprovine@stanford.edu; sgchong@stanford.edu; jwatkins@stanford.edu; rthowe@stanford.edu; hspwong@stanford.edu; subh@stanford.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2012.2232707

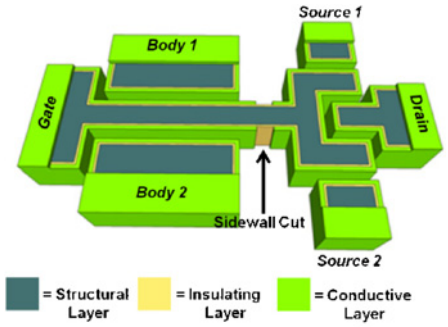


Fig. 2. 3-D schematic of a lateral six-terminal relay with a Y-shaped cantilever beam.

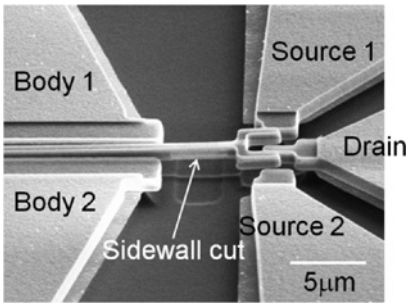


Fig. 3. Oblique SEM of a completed lateral six-terminal relay with a Y-shaped movable gate. The minimum feature size is 500 nm, which is set by the lithographic resolution of *i*-line optical photolithography.

In this paper, we present techniques based on binary decision diagrams (BDDs) to design NEM-relay based logic using six-terminal NEM relays. In Section II, we show that a six-terminal relay can be used as a 2-to-1 multiplexer (MUX) when properly biased, which enables the relay to be used as a universal logic gate. In Section III, we implement arbitrary combinational logic circuits with a single mechanical delay using well-known BDD techniques. We also outline a method for designing datapath circuits using cascaded, BDD-based subblocks. For complex functions, implementations with a single mechanical delay can require a prohibitively large number of relays. In Section IV, we present two examples of complex circuits to show that the total number of relays can be reduced at the expense of additional mechanical delays. We present an overview of six-terminal relays in Appendices I–IV.

II. Six-terminal Relay as a Logic Element

A. Device Structure and Operation

Fig. 2 shows a schematic of the top view of a six-terminal relay consisting of a gate (G), a drain (D), two bodies (B_1, B_2), and two sources (S_1, S_2). The movable part of the gate electrode is a Y-shaped cantilever beam that is laterally and bi-directionally actuated (i.e., the cantilever beam moves in-plane to either side). The electrodes consist of a patterned structural layer [e.g., polysilicon deposited by chemical vapor deposition (CVD)] on an insulating layer [e.g., low temperature oxide (LTO) deposited by CVD] that isolates the electrodes from the substrate (e.g., silicon). All the electrodes have an insulating layer on the sidewalls [e.g., hafnium oxide deposited by atomic layer deposition (ALD)] that provides electrical isolation between the sidewalls and the interior of the electrodes. A conductive sidewall [e.g., titanium nitride deposited by metal organic chemical vapor deposition (MOCVD)] is fabricated on the outside of the insulating layer and is selectively removed yielding a conductive channel on the beam that is electrically isolated from the gate electrode. The conductive sidewall extends to the surface of the electrodes in the pad areas to create contacts for electrical probing. The Y-shaped beam is released (i.e., separated) from the substrate by a timed isotropic etching of the underlying insulating layer followed by critical point drying to reduce stiction. Fig. 3 shows an oblique SEM of the fabricated device where LTO, polysilicon, hafnium oxide, and titanium nitride are utilized for the insulating layer, the structural layer, the insulating sidewall, and the conductive sidewall, respectively. Appendix I and [7] discuss the process flow in more detail. The fabrication process can be adapted to other choices of materials.

The Y-shaped beam can be actuated bidirectionally toward B_1 and B_2. Assume that the G and B_2 electrodes are grounded, while the B_1 voltage is swept. When the B_1 voltage is zero, the beam is not deflected and S_1 and S_2 are mechanically and electrically isolated from D. As the B_1 voltage increases, the beam moves toward B_1 due to the increased electrostatic force between the beam and B_1 that is balanced by the elastic force of the beam. Ultimately, as the B_1 voltage continues to increase beyond the voltage called the pull-in voltage, the electrostatic force increases faster than the elastic force, and the beam collapses into S_1 and D. At this critical point, D is electrically connected to S_1 via the channel sidewalls. The B_1 voltage is then decreased which decreases the electrostatic force between the beam and B_1. Due to the reduced gap between the beam and B_1, the B_1 voltage must be reduced below the pull-in voltage before the elastic force can balance the electrostatic force. At this point, the beam comes out of contact with S_1 and D, and S_1 and D are again isolated. This voltage is called the pull-out voltage. The device works similarly under the actuation toward B_2.

Fig. 4 shows the measured electrical characteristic for a similar relay as shown in Fig. 3, in which the B_1 voltage is swept, while the G and B_2 are grounded. The pull-in and pull-out voltage of the relay are measured to be 31 and 21.6 V, where the beam length, beam thickness, initial beam-to-body gap, and initial beam-to-drain gap of the relay are 20 μm, 660 nm, 440 nm, and 340 nm, respectively. In the off-state, no current flows through the device. In the on-state, the current only flows between S_1 and D with zero leakage between G, B_1, and B_2, confirming the correct operation with electrical isolation between G and the channel sidewalls.

B. Six-Terminal Relay as a 2-to-1 Multiplexer

First, we show that a 2-to-1 MUX can be built using a six-terminal relay. The electrical configuration is shown in Fig. 5(a). Body electrodes B_1 and B_2 are connected to V_{BB} and GND, respectively, where V_{BB} is greater than the pull-in voltage for a single-sided actuation. When the gate voltage

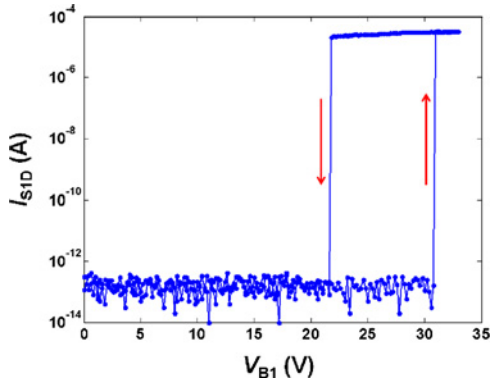


Fig. 4. Current–voltage characteristic of a six-terminal relay. The gate, body 2, and source 2 are connected to GND. Source 1 and drain are connected to 3 and −3 V, respectively, while body 1 was double swept between 0 and 33 V.

is GND, the beam is pulled in toward B_1, and output (D) is connected to S_1. When the gate voltage is V_{BB}, the beam is pulled in toward B_2 and the output is connected to S_2.

By connecting digital inputs to S_1 and S_2, a compact 2-to-1 MUX is constructed. The output (D) is expressed as

$$D = \bar{G}S_1 + GS_2. \tag{1}$$

Due to the electrical isolation between the channel sidewalls and the gate electrode, the voltages at S_1 and S_2 can be chosen independently from the voltages at B_1, B_2, and G. However, if we need to construct a cascadable circuit by connecting the output of a 2-to-1 MUX to the gate input of another 2-to-1 MUX, the voltage range at D must match the voltage range at G.

Alternatively, a 2-to-1 MUX can be formed by connecting two gates and two drains of two four-terminal relays as shown in Fig. 5(b), which enables the design to be implemented even if only four-terminal devices are available. The logic symbol of a 2-to-1 MUX is shown in Fig. 5(c).

A comparison between implementing the 2-to-1 MUX using a single six-terminal relay versus two four-terminal relays is presented in Appendix II. We show that the six-terminal relay implementation has inherent advantages compared to the four-terminal relay implementation including a reduced footprint and no potential hazard (i.e., no direct conductive path between two sources). In Appendix III, we present a device model of a laterally actuated six-terminal relay and discuss device scaling. Appendix IV shows the advantages of using laterally actuated relays compared to vertically actuated relays.

III. Single Mechanical Delay Implementation of Arbitrary Combinational Logic Using Six-Terminal Relays

A. Implementation Using Binary Decision Diagrams

Any arbitrary combinational logic function can be represented by a BDD where each node is mapped to a 2-to-1 MUX [8], [9]. Since a 2-to-1 MUX can be implemented using a six-terminal relay, any arbitrary combinational logic function can be implemented using a six-terminal relay.

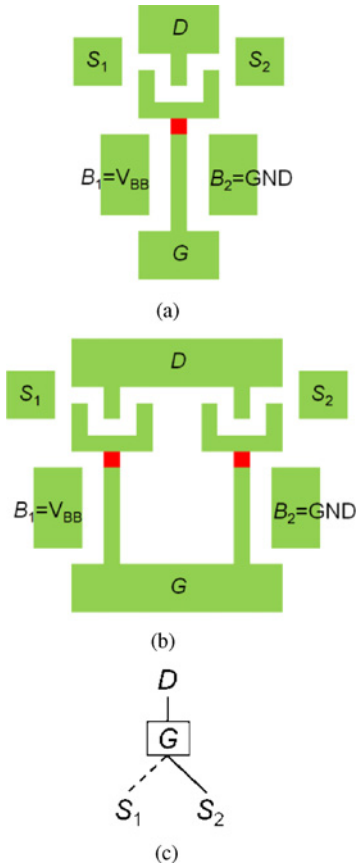


Fig. 5. Electrical configurations for a 2-to-1 MUX using: (a) one lateral six-terminal relay, (b) two lateral four-terminal relays with connected drain and gate. The beam isolation regions are shown in red. (c) Logic symbol of a 2-to-1 MUX.

For any arbitrary combinational logic function, the states of all six-terminal relays in the corresponding BDD representation are determined simultaneously after one mechanical delay as long as the gate input of each six-terminal NEM relay is connected to a primary input. Terminal 1's and terminal 0's are connected to V_{DD} and GND, respectively. Fig. 6 shows two BDD implementations of $F = AB + CD$; the first implementation connects all the source terminals to V_{DD} and GND [Fig. 6(a)] and the second implementation eliminates one relay by connecting the signal D directly to a source [Fig. 6(b)]. After one mechanical delay, one conducting path is established from the source of one of the NEM relays (connected to a primary input, V_{DD}, or GND) to the output. Once this conduction path is established, there is an electrical delay associated with the time required to charge or discharge all the nodes on the conduction path. Each node of the BDD has an associated contact resistance and parasitic capacitance. Therefore, the electrical delay required to propagate from the bottom to the top of the BDD network is a quadratic function of the number of six-terminal relays on the path. The total switching time is the sum of one mechanical delay and the electrical delay along that path. The electrical delay is generally much less than the mechanical delay. Therefore, the switching time for the BDD circuit can be approximated as one mechanical delay unless there is a large electrical

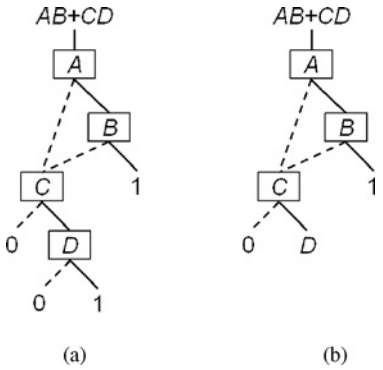


Fig. 6. Single mechanical delay implementation of $F = AB + CD$. (a) Sources of the relays are connected to V_{DD} and GND only. (b) Sources of the relays may be connected to primary inputs in addition to V_{DD} and GND; this can reduce the number of relays.

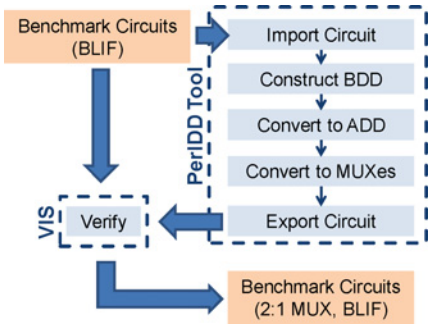


Fig. 7. Design flow using PerlDD and VIS tools. PerlDD automates the generation of BDD circuits with all nodes being driven by primary inputs and verification is performed by VIS.

TABLE I
TOTAL NUMBER OF RELAYS, THE NUMBER OF RELAYS ON THE CRITICAL PATH, AND RUNTIME FOR MCNC BENCHMARK CIRCUITS

Circuit Name	Number of Relays	Number of Relays on the Critical Path	Runtime (ms)
alu4	775	14	220
apex4	1108	9	182
des	4006	18	740
ex1010	1116	11	628
ex5p	308	8	178
misex3	657	14	212
pdc	2474	16	686
seq	1978	23	296
spla	847	16	566
8-b adder	147	17	52
16-b adder	182	33	70
8 × 8 multiplier	13 475	16	748

delay due to many six-terminal relays in the conduction path. The six-terminal relays in this paper were fabricated in a 1 μm process and have a parasitic capacitance, $C_{\text{sw-b}}$, between the isolated conductive sidewall and the conductive beam that is approximately 180 fF, a contact resistance, R_{con}, of approximately 1 kΩ, and a mechanical delay, t_{mech}, of approximately 1 μs. Using an Elmore delay estimation, the electrical delay will be comparable to the mechanical delay when there are 125 relays in the conduction path. This is comparable to prior works based on four-terminal relays that showed that there must be on the order of hundreds of relays in the conduction path to cause the electrical delay to be equal to the mechanical delay [4], [5]. For six-terminal relays scaled to a 22 nm node with a mechanical delay of 10 ns, 63 relays in the conduction path would make the electrical delay comparable to the mechanical delay. This number could be increased to several hundred relays if the parasitic capacitance is reduced by making the beam material nonconductive.

To build the BDD for any combinational circuit, we utilize the PerlDD program, which is a Perl extension of CUDD, to generate BDD designs as shown in Fig. 7 [10]. We first import a combinational circuit in BLIF format. PerlDD constructs a BDD network for each primary output of the circuit. PerlDD constructs a BDD network for each primary output of the circuit and automatically handles variable reordering. The tool generates BDD networks that use some BDD nodes in complement form, which require an extra mechanical delay when implemented with NEM relays. This problem is solved by converting the BDD to an algebraic decision diagram (ADD) with 0 and 1 terminals before mapping the circuit to 2:1 MUXes. The circuit is exported into BLIF and verified against the original circuit using the verification interacting with synthesis (VIS) tool [11]. Table I shows the number of nodes required for the single mechanical delay implementation of various MCNC benchmark circuits. The logic synthesis was run on an AMD Phenom II X4 955 processor, which is a 3.20 GHz quad core processor with 64 kB L1 cache and 512 kB L2 with 6 GB of memory.

Although the PerlDD flow discussed above can generate any combinational circuit, the BDD implementation does not necessarily produce the most compact MUX-based circuit with the fewest nodes. For example, combinational logic circuits with well-partitioned logic blocks, e.g., bit-sliced datapath circuits, can be constructed by cascading BDD sublogic blocks (as shown in Fig. 8). A single mechanical delay implementation is obtained by connecting all the gate terminals to primary inputs, and the outputs of the sublogic blocks are only connected to the source terminals of the six-terminal relays. As long as these constraints are satisfied, the design will still require only a single mechanical delay even if there are carry signals arriving from multiple blocks.

To illustrate this, we consider an example of an n-bit ripple carry adder. The expressions for the carry (C_i) and the sum (S_i) outputs of the ith stage are given by

$$C_i = A_i B_i + C_{i-1}(A_i + B_i) \tag{2}$$

$$S_i = A_i \oplus B_i \oplus C_{i-1}. \tag{3}$$

The ripple carry adder can be implemented with a single mechanical delay in the following way: for each bit-slice i, create BDDs for S_i, C_i, and $\bar{C}_i$ such that the gates of the relays are connected to A_i and B_i only as shown in Fig. 9. C_{i-1} and $\bar{C}_{i-1}$ are connected to source inputs so that they do not incur additional mechanical delays. Both C_i and $\bar{C}_i$ must be calculated simultaneously by individual logic blocks. Otherwise, an additional mechanical delay would be required

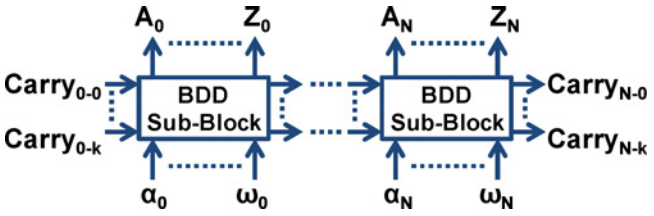


Fig. 8. Single mechanical delay implementation of N-bit datapath circuit using cascaded BDD, sublogic blocks with an arbitrary number of carry, input, and output signals. To maintain a single mechanical delay, all the gates must be connected to primary inputs, α to ω, and all carry signals are connected to the source terminals of the six-terminal relays.

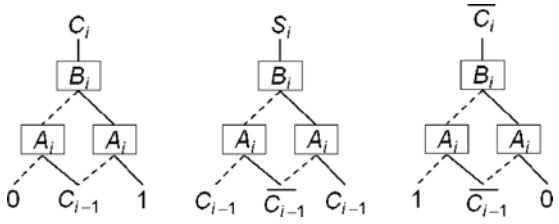


Fig. 9. Relays required to implement adder subblock. BDDs are formed to calculate S_i, C_i, and $\overline{C_i}$ simultaneously.

to invert the signal. The complete circuits for S_i, C_i, and $\overline{C_i}$ are constructed by cascading circuits for each bit slice. As shown in Fig. 9, the adder subblock requires a total of 9 six-terminal relays. An n-bit adder is formed by cascading the adder subblocks together as shown in Fig. 8 and the adder requires $9n$ relays. Cascading subblocks based on BDDs can be done for any iterative network with a single carry. The adder example shows that even multiple carries can be handled; both the carry signal and its complement are calculated and propagated (as a result, the number of relays required can grow significantly). In comparison to the adder implementations designed using PerlDD, the bit-slice technique reduces the number of relays from 147 to 72 for an 8-b adder and from 182 to 144 for an 16-b adder.

B. Comparison With CMOS-Style Implementation

A four-terminal relay can be used to implement any arbitrary combinational logic function with a single mechanical delay [4]. However, the number of six-terminal relays required for the BDD-based implementation of a combinational logic function is less than or equal to half of the number of four-terminal relays for a CMOS-style implementation. In a CMOS-style implementation, four-terminal relays are biased so that they perform similarly to NMOS and PMOS transistors (i.e., for high body bias voltages, low gate voltages lead to a low resistance between the source and drain and conversely, for low body bias voltages, high gate voltages lead to a low resistance between the source and drain). These relays are connected in series or parallel combinations thereof to form pull-up and pull-down networks; no bridge connections (i.e., transistors that connect two branches of the pull-up or pull-down network as used in some adder implementations [12]) are permitted.

As discussed in Appendix II, a 2-to-1 MUX can be built with smaller area when using a single six-terminal relay

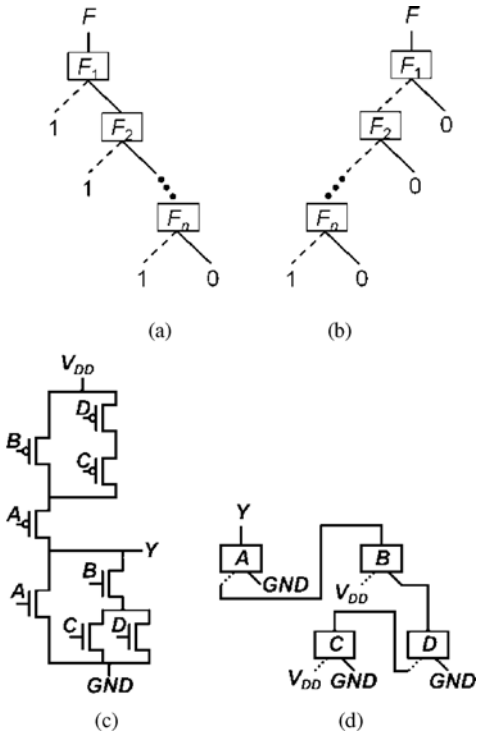


Fig. 10. BDD construction using BDDs. (a) n-input NAND. (b) n-input NOR. F_i for $1 \leq i \leq n$ denotes the BDD. (c) and (d) CMOS-gate and BDD implementations, respectively, for $Y = \overline{A + B \cdot (C + D)}$. The BDD implementation only requires four six-terminal relays compared to eight gates for the CMOS-style implementation.

instead of using two four-terminal relays, suggesting overall area benefits when using six-terminal relays instead of four-terminal relays.

Consider the pull-up network of an n-input NAND gate. There are n parallel four-terminal relays. Due to duality, the pull-down network needs n four-terminal relays in series. Thus, a total of $2n$ four-terminal relays are required for the CMOS-style implementation. The BDD of the NAND gate will have n nodes, thus, the BDD implementation requires n six-terminal relays. Similarly, for an n-input NOR gate with n series four-terminal relays for its pull-up network, the number of four-terminal relays for the CMOS-style implementation is twice the number of six-terminal relays in the BDD implementation. Fig. 10 shows how to arrange BDDs to construct BDDs for n-input NAND [Fig. 10(a)] and n-input NOR [Fig. 10(b)].

Any arbitrary CMOS network can be converted to a relay-based implementation by recursively converting its serial and parallel branches into their equivalent relay-based implementations. To convert an arbitrary CMOS network to a MUX network, each NMOS transistor of the pull-down network is mapped to a MUX network consisting of a six-terminal relay driven by the same gate input as the NMOS transistor. Logic-zero and logic-one terminals are defined as the source connections corresponding to S_1 and S_2 as shown in Fig. 5(a). If there are two MUX networks, f_1 and f_2, connected in series, then the logic-one terminal of f_1 will be replaced by the output terminal of f_2. Alternatively, if f_1 and f_2 are connected in parallel, then the logic-zero terminal of f_1 will be replaced by the output of f_2. For both serial and parallel connections,

all logic-zero and all logic-one terminals are merged into a single logic-zero and a single logic-one terminal for the merged network. Once all the individual MUXes have been merged, the pull-down network is completed by connecting the logic-one terminal of the final network to GND, and the pull-up network is completed by connecting logic-zero to V_{DD}. The MUX network and CMOS network implementations require the same number of relays or transistors, respectively, to implement the pull-down network.

Fig. 10(c) and (d) shows an example of how a circuit with both serial and parallel connections is mapped to six-terminal relays; the MUX network implementation requires four six-terminal relays compared to eight CMOS transistors for the CMOS-style implementation. This mapping strategy is useful for converting existing CMOS-style implementations to MUX networks and for demonstrating that the MUX implementation requires one half of the transistors compared to the CMOS-style implementation. In practice, PerlDD as described in Section II-A or another BDD program would be used to generate the BDD implementation directly from the required logic function, which can take advantage of variable reordering to further improve the BDD implementation.

Note that the BDD obtained from this mapping may not be a reduced ordered binary decision diagram; there might be some redundant nodes, some variables may appear more than once on a path, and different paths may have different variable ordering, suggesting the possibility of further optimization. To illustrate the first point, consider the following function, $F = AG + \bar{A}H$. A CMOS-style implementation will require eight four-terminal relays, thus, a BDD-based implementation constructed from the CMOS-style implementation will require four six-terminal relays. The function F can be implemented with a single six-terminal relay by connecting the signal lines, G and H, directly to the sources of the six-terminal relay.

In CMOS, there are several drawbacks to pass transistor logic (PTL) or transmission gate logic such as V_{TH} voltage drops for PTL, dual rail signaling for the control lines, additional wiring complexity, input/output coupling, lack of signal restoration, and added signal line capacitance [13]. For six-terminal relays, there is no V_{TH} voltage drop and dual rail signaling is not required, which reduces the wiring complexity. Input/output coupling is avoided by biasing the six-terminal relay so that the drain of the relay is always connected to either S_1 or S_2 (i.e., the drain never remains in a high impedance state after switching). Signal restoration is not expected to be any worse than the CMOS-style implementation as long as sufficient time is allowed for all the capacitances in the circuit to charge. However, six-terminal relays will still suffer from added signal line capacitance for designs where the signal lines are connected to the source terminals.

Finally, we illustrate the difference between the electrical critical paths of two implementations for $F = AB + CD$. The number of six-terminal relays for the BDD-based implementation is 3 [Fig. 6(b)], while the number of four-terminal relays for the CMOS-style implementation is 8. However, the BDD-based implementation has three relays on its critical path [Fig. 6(b)], while the CMOS-style implementation has 2 relays on its critical path. Thus, for some functions, the CMOS-style implementation has a smaller number of relays on its critical path compared to the BDD-based implementation. Unless the number of relays in a conduction path is significantly high, the impact will not be significant since the switching time is dominated by the mechanical delay.

IV. Multiple Mechanical Delay Implementation

In Section III, we have shown that arbitrary combinational logic functions can be efficiently implemented with a single mechanical delay using the BDD-based representation. We have also found that when there are many relays in series in the conduction path, the electrical delay can surpass the mechanical delay and dominate the switching time [4]. Furthermore, the number of relays may be prohibitively large for some functions such as multipliers [14] since the BDD size can be very large for such functions. Allowing multiple mechanical delay implementations where intermediate inputs can drive the gate electrodes can reduce the total delay or reduce the total number of relays required in these situations.

There have been some studies in performing multilevel logic conversion through BDD decomposition [15]. The decomposition points called dominators are found by scanning the BDD [15]. In a multilevel BDD, the gate inputs of the BDDs at each level are connected not only to the primary inputs, but also to the outputs of other BDDs. BDD decomposition techniques can be used to find a multilevel BDD representation of any arbitrary function and use it to implement a multiple mechanical delay circuit. BDD decomposition is a computationally expensive operation.

In Sections IV-A and IV-B, we present two examples: a datapath circuit with two carry signals and an 8×8-b array multiplier. These examples show tradeoffs between the number of relays required to implement a circuit and the total number of mechanical delays.

A. Example: Datapath Circuit With Two Carry Signals

Consider an extension of the ripple carry adder of Section III-A. The ith bit slice ($1 \leq i \leq n$) has two carry signals S_i and T_i expressed as

$$S_i = A_i\bar{S}_{i-1}\bar{T}_{i-1} + B_i\bar{S}_{i-1}T_{i-1} + C_iS_{i-1}\bar{T}_{i-1} + D_iS_{i-1}T_{i-1} \quad (4)$$

$$T_i = E_i\bar{S}_{i-1}\bar{T}_{i-1} + F_i\bar{S}_{i-1}T_{i-1} + G_iS_{i-1}\bar{T}_{i-1} + H_iS_{i-1}T_{i-1} \quad (5)$$

where A_i, B_i, C_i, D_i, E_i, F_i, G_i, and H_i are primary inputs in the ith bit slice. The outputs are carry signals.

There are two approaches for designing a single mechanical delay circuit. In the first approach, we find the BDDs of each bit slice where all the gates of the six-terminal relays are connected to primary inputs and cascade them to build BDDs. This is similar to the ripple carry adder example in Section III-A. As shown in Fig. 11, all 16 Boolean functions of two carry signals, S_{i-1} and T_{i-1}, are required at the source inputs of the bottom relays to implement the BDDs of S_i and T_i. Therefore, to maintain a single mechanical delay implementation, the BDDs of 12 new carry signals (i.e., excluding S_i, T_i, 1, and 0)

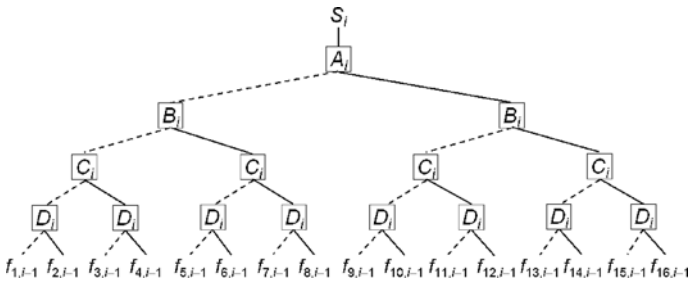


Fig. 11. First approach for single mechanical delay implementation of bit slices with two carry signals. The BDD of S_i for the ith bit slice is shown where the gates are connected to primary inputs. The source inputs consist of all 16 Boolean functions of two carry signals, S_{i-1} and T_{i-1}, denoted by $f_{1,i-1} \sim f_{16,i-1}$ (see Table II for definitions).

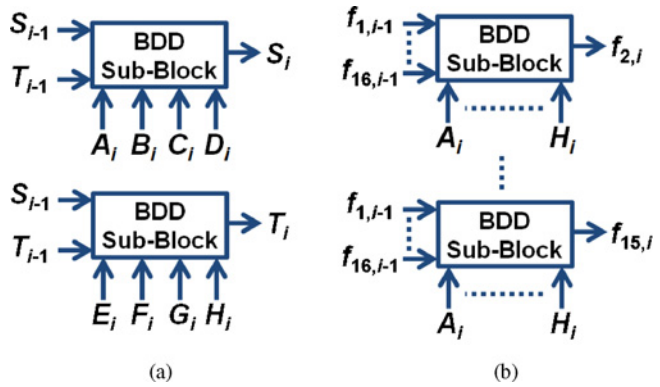


Fig. 12. (a) Block diagrams of the ith bit slice for implementing two carry signals S_i and T_i. (b) Actual block diagrams for a single mechanical delay implementation. All Boolean functions of two carry signals should be generated (see Table II for definitions of $f_{1,i} \sim f_{16,i}$).

must be generated as shown in Fig. 12. Table II shows the number of nodes per bit-slice that are required to generate all the Boolean functions for generating the carry signals. Therefore, a total of $390n$ relays are required for the first approach.

In the second approach, we build BDDs of S_i and T_i for all i's by directly expressing S_i and T_i in terms of primary inputs only. Fig. 13 shows the BDDs of S_1 and S_2. The BDD of S_i can be created from that of S_{i-1} by replacing the sources, A_{i-1}, B_{i-1}, C_{i-1}, and D_{i-1}: A_{i-1}, B_{i-1}, C_{i-1}, and D_{i-1} are replaced with

$$A_i\bar{A}_{i-1}\bar{E}_{i-1}+B_i\bar{A}_{i-1}E_{i-1}+C_iA_{i-1}\bar{E}_{i-1}+D_iA_{i-1}E_{i-1},$$
$$A_i\bar{B}_{i-1}\bar{F}_{i-1}+B_i\bar{B}_{i-1}F_{i-1}+C_iB_{i-1}\bar{F}_{i-1}+D_iB_{i-1}F_{i-1},$$
$$A_i\bar{C}_{i-1}\bar{G}_{i-1}+B_i\bar{C}_{i-1}G_{i-1}+C_iC_{i-1}\bar{G}_{i-1}+D_iC_{i-1}G_{i-1},$$
$$\text{and } A_i\bar{D}_{i-1}\bar{H}_{i-1}+B_i\bar{D}_{i-1}H_{i-1}+C_iD_{i-1}\bar{H}_{i-1}+D_iD_{i-1}H_{i-1}$$

respectively. This step adds 12 relays [i.e., num(S_i) = num(S_{i-1})+12 where num(f) is the number of six-terminal relays required to implement the BDD of f]. Since num(S_1) = 3, num(S_i) = $12i-9$. Similarly, the BDD of T_i can be created from that of T_{i-1}, yielding num(T_i) = num(S_i) = $12i-9$. The total number of relays for creating carry signals for all n stages is $\sum_{i=1}^{n}(\text{num}(S_i)+\text{num}(T_i)) = n(12n-6)$. This second approach uses fewer relays compared to the first approach for $n \leq 33$. The number of relays on the critical path for S_n and T_n of the first approach and the second approach is $8n$ and $2n$, respectively.

TABLE II
TWO-INPUT (S_i AND T_i) BOOLEAN FUNCTIONS GENERATED FOR THE SOURCE INPUTS OF THE CASCADED BIT-SLICE IMPLEMENTATION (APPROACH 1)

Boolean Function	Relays Required to Implement the Function per Bit-Slice	Number of Relays on the Critical Path
$f_{1,i}=0$	0	0
$f_{2,i}=S_i\bar{T}_i$	30	8
$f_{3,i}=S_i\bar{T}_i$	30	8
$f_{4,i}=S_i$	15	4
$f_{5,i}=\bar{S}_iT_i$	30	8
$f_{6,i}=T_i$	15	4
$f_{7,i}=S_i\oplus T_i$	45	8
$f_{8,i}=S_i+T_i$	30	8
$f_{9,i}=S_i\bar{T}_i$	30	8
$f_{10,i}=S_i\oplus\bar{T}_i$	45	8
$f_{11,i}=\bar{T}_i$	15	4
$f_{12,i}=S_i+\bar{T}_i$	30	8
$f_{13,i}=\bar{S}_i$	15	4
$f_{14,i}=\bar{S}_i+T_i$	30	8
$f_{15,i}=\bar{S}_i+\bar{T}_i$	30	8
$f_{16,i}=1$	0	0

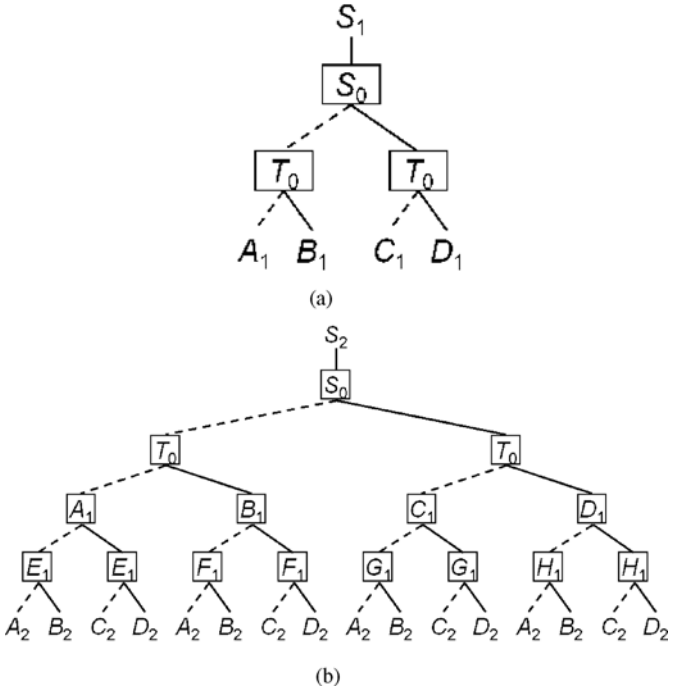


Fig. 13. Second approach for single mechanical delay implementation of bit slices with two carry signals. The BDDs of all carry signals are minimized separately. (a) BDD of S_1. (b) BDD of S_2.

Both approaches discussed above require many relays. The number of relays required to implement the above function can be significantly reduced by allowing multiple mechanical delays. Fig. 14 implements the same circuit by connecting the carry signals to the gates of the six-terminal relays. This implementation requires n mechanical delays but it only requires $6n$ ($3n$ each for S_i and T_i) relays.

Tradeoff between the number of relays and the number of mechanical delays can be achieved in the following way: n-bit slices are divided into groups, each consisting of k-bit

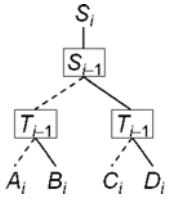


Fig. 14. n mechanical delay implementation for bit slices with two carry signals. The BDD of S_i for the ith bit slice is shown where the gate inputs are S_{i-1} and T_{i-1}.

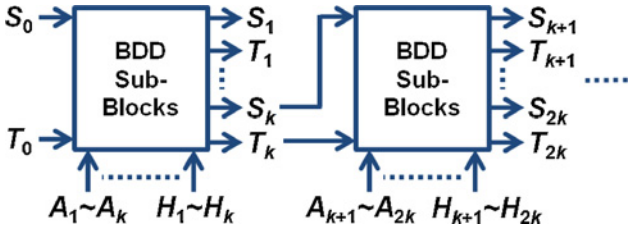


Fig. 15. n/k mechanical delay implementation for bit slices with two carry signals. BDD subblocks correspond to k bit slices that are implemented with a single mechanical delay using the second approach (Fig. 13). S_{ik} and T_{ik} $(0 \le i \le n/k - 1)$ are treated as primary inputs (i.e., connected to the gate terminals). For simplicity, we assume that n is the multiple of k.

slices as shown in Fig. 15. The BDDs for implementing each group are created using the second approach for the single mechanical delay implementation. The total number of relays will be $(n/k)k(12k - 6) = n(12k - 6)$, whereas the number of mechanical delays will be n/k.

B. Example: 8 × 8-B Array Multiplier

Consider an 8 × 8 array multiplier that is implemented using a carry-save adder tree followed by a ripple carry adder [16]. The multiplicand is $A = \sum_{i=0}^{7} A_i 2^i$ and the multiplier is $B = \sum_{i=0}^{7} B_i 2^i$. We first need to generate eight partial products that are defined as $P_i = B_i \cdot A \cdot 2^i (0 \le i \le 7)$. Since each partial product can be generated using 8 relays in a single mechanical delay, this step can be implemented in a single mechanical delay using 64 relays. Next, we use a carry-save adder tree to reduce the sum of the 8 partial products to the sum of two numbers followed by a ripple carry adder for obtaining the final sum as shown in Fig. 16. Since there are four stages in the carry-save adder tree, the reduction step can be implemented using 4 mechanical delays. The relay count for the reduction step is 352, which is calculated by counting the number of full-adders and half-adders. A full-adder and a half-adder can be implemented using 8 and 4 relays, respectively, as shown in Fig. 17. The final sum is performed using a 13-b ripple carry adder, implemented using 117 relays in a single mechanical delay. Thus, the multiplier has 6 mechanical delays and 533 relays.

The stages in the carry-save adder tree can be merged to reduce the number of mechanical delays at the expense of increased relay count. Fig. 18 shows a single mechanical delay implementation for the reduction step. Here, all intermediate outputs (i.e., carryouts and sums) should be implemented in a single mechanical delay, whose BDDs can be obtained recursively using the BDDs of operands for a full-adder and a half-adder as shown in Fig. 19. The number of relays

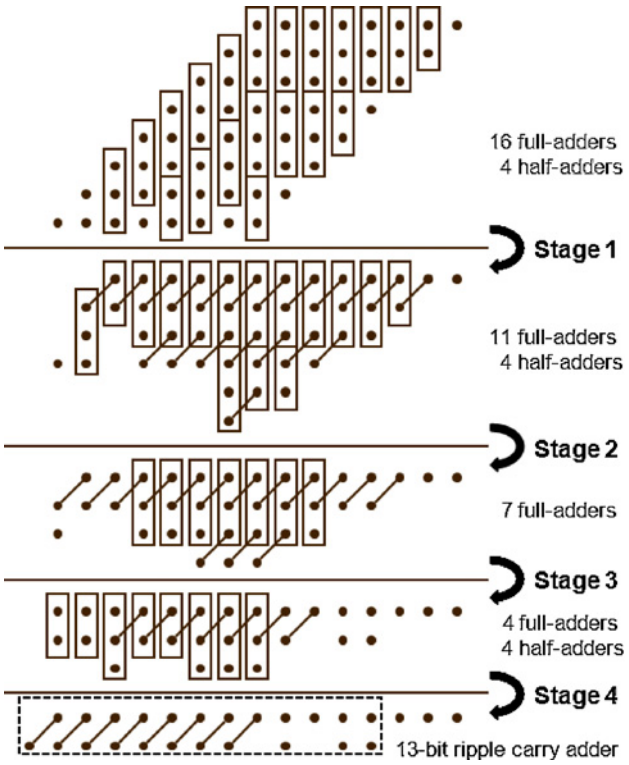


Fig. 16. 8 × 8 multiplier implementation with 6 mechanical delays using a carry-save adder tree followed by a ripple carry adder. One mechanical delay is required for generating eight partial products, and 4 mechanical delays are required for reducing the 8 partial products to two numbers using four stages of a carry-save adder tree, and 1 mechanical delay is required for obtaining a final sum. Boxes with three dots and two dots correspond to full-adders and half-adders, respectively. Two dots that are connected with a 45° line are carryout and sum from an adder (see the notations in Fig. 17).

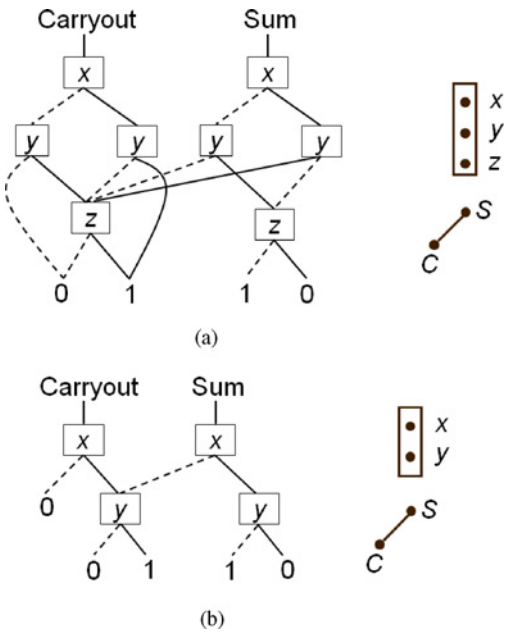


Fig. 17. Carry-save adder implementation. (a) BDDs for a full-adder and its notation in the carry-save adder tree. (b) BDDs for a half-adder and its notation in the carry-save adder tree. x, y, and z are primary inputs. The carryout is $C = xy + yz + zx$ and the sum is $S = x \oplus y \oplus z$.

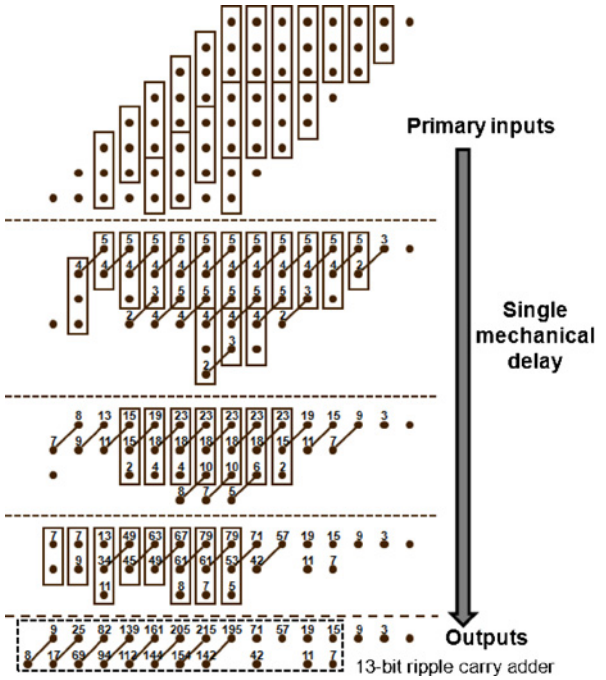


Fig. 18. 8×8 multiplier implementation with 3 mechanical delays using the same approach as in Fig. 16, except that the reduction step is carried out with a single mechanical delay. For this, all intermediate outputs in the reduction step should be implemented in a single mechanical delay, whose BDDs can be obtained recursively using the BDDs for a full-adder and a half-adder as shown in Fig. 19. The number of relays for the intermediate outputs (corresponding to the number on top of each dot) can be counted recursively. The number of relays required at this reduction step is calculated by summing the numbers at the final sum step.

required to implement the intermediate outputs (corresponding to the number on top of each dot in Fig. 18) can be counted recursively—for a full-adder that adds f, g, and h, num(carryout) = num(f) + 2num(g) + num(h) and num(sum) = num(f) + 2num(g) + 2num(h). For a half-adder that adds f and g, num(carryout) = num(f) + num(g) and num(sum) = num(f) + 2 num(g). This results in a multiplier implementation with 3 mechanical delays and 2186 relays. A 4 mechanical delay implementation is obtained by merging the stage 2, 3, and 4 only. Here, the number of relays for the reduction step is the sum of the number for this merge and the number for stage 1. A 5 mechanical delay implementation can be obtained similarly. For 1 or 2 mechanical delay implementation, the three steps should be merged further in a similar manner. Table III summarizes the 8×8 multiplier implementation using this approach. For comparison, the number of four-terminal relays for CMOS-style implementation using the same partitions is included in Table III. In this example, we observe that as the number of mechanical delays decreases, the ratio of the number of six-terminal relays to that of four-terminal relays decreases due to better sharing of subfunctions for BDD-based implementation.

V. Conclusion

In this paper, we presented techniques for implementing arbitrary combinational logic functions based on universal logic gates using six-terminal NEM relays. The six-terminal relay was shown to function as a 2-to-1 MUX with appropriate biasing. The delays of the resulting circuits were dominated by the mechanical delays of the relays unless there are many relays (more than hundreds) in series in the conduction path. In this case, the total number of relays can be reduced at the expense of additional mechanical delays. Allowing multiple mechanical delay implementations can also reduce the total delay for circuits where the delays are dominated by the electrical delay. Future work includes: 1) algorithms for optimizing multiple mechanical delay implementations for any arbitrary combinational logic function; 2) experimental demonstration and characterization of six-terminal relay-based circuits designed using our methodologies; and 3) development of highly optimized macro blocks (potentially optimized using techniques beyond the BDD-based approaches described in this paper) and design methodologies to use them for overall timing and area optimization.

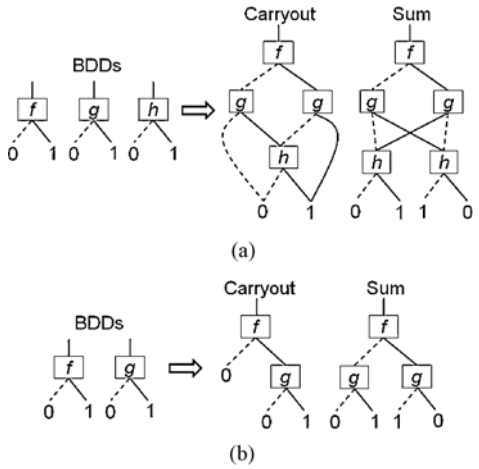


Fig. 19. BDD implementations of carryout and sum for (a) a full-adder, (b) a half-adder using the BDDs of operands, f, g, and h. The implementations do not add additional mechanical delays.

TABLE III
Tradeoff Between the Number of Mechanical Delays and the Number of Six-Terminal and Four-Terminal Relays for an 8×8 Multiplier Using a Carry-Save Adder Tree Followed by a Ripple Carry Adder

Number of Mechanical Delays	Number of Six-Terminal Relays in BDD-Based Implementation	Number of Four-Terminal Relays in CMOS-Style Implementation	Ratio
1	16 352	270 284	0.06
2	4129	45 456	0.09
3	2186	23 138	0.09
4	875	5002	0.17
5	590	2214	0.27
6	533	1852	0.29

Appendix I
Fabrication Process

A three-mask process, illustrated by the cross-sections in Fig. 20, was used to fabricate six-terminal, laterally actuated relays. In this process, the structural layer is doped polysilicon,

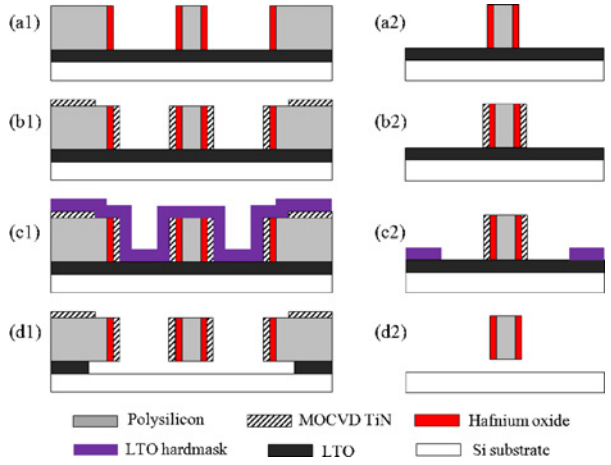


Fig. 20. Fabrication process of lateral six-terminal relays. (a) Polysilicon patterning (Mask 1) followed by hafnium oxide sidewall formation by a blanket RIE. (b) MOCVD TiN deposition and patterning with RIE (Mask 2). (c) LTO hardmask deposition and patterning with an isotropic wet etch (6:1 BOE) (Mask 3). (d) Isotropic wet etch of the exposed TiN sidewall and release of the devices in 6:1 BOE followed by critical point drying. (a1-d1) shows a cross-section that does not contain the isolation region whereas (a2-d2) shows a cross-section along the isolation region.

while the insulating and conductive sidewalls are made of hafnium oxide (HfO_2) and titanium nitride (TiN), respectively. The process starts with deposition of 1.5 μm of low-temperature, chemical-vapor deposited silicon dioxide (LTO) and a 1-μm-thick layer of phosphorus-doped polysilicon. The polysilicon is patterned (Mask 1) using i-line optical photolithography and a reactive-ion-etch (RIE). Second, a 33 nm-thick hafnium oxide is deposited via ALD on a Cambridge Nanotech Savannah S200 followed by a blanket RIE, leaving hafnium oxide only on the sidewalls [Fig. 20(a)]. Third, a 50-nm-thick TiN layer is conformally deposited using MOCVD [17] and directionally etched to clear the top and the bottom of trenches, except the pad areas protected by Mask 2 [(Fig. 20(b)]. Fourth, a 120-nm-thick LTO layer is deposited and the third lithographic step (Mask 3) is performed to create regions where the LTO is selectively etched to access the TiN sidewall. The exposed masking LTO is isotropically etched in 6:1 buffered oxide etch (BOE) and then the photoresist mask is stripped [Fig. 20(c)]. Next, the exposed TiN sidewall is isotropically etched in NH_4OH:H_2O_2:H_2O = 1:2:5 at 60 °C using a timed etch. This solution negligibly etches LTO, hafnium oxide, and polysilicon [18]. This etch step leaves TiN in the areas of contact and removes TiN from part of the cantilever to electrically isolate the contact region of the actuation electrode. Lastly, the devices are released in 6:1 BOE by another timed etch that etches the masking and sacrificial LTO layers. The final step is critical point drying [Fig. 20(d)]. This three-mask process is self-aligned and enables a compact and symmetric, lateral six-terminal relay.

APPENDIX II
COMPARISON BETWEEN THE TWO 2-TO-1 MUX IMPLEMENTATIONS

Implementing a 2-to-1 MUX using one six-terminal relay has many inherent advantages over using two four-terminal relays. First, the area is reduced when using one six-terminal

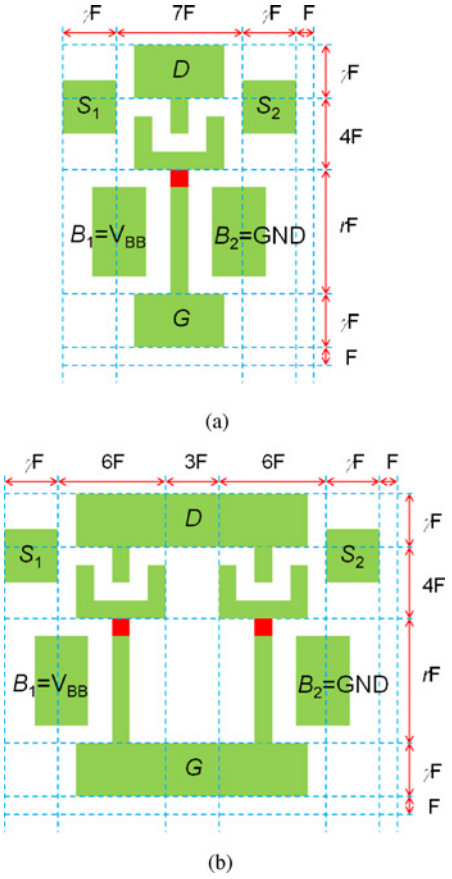


Fig. 21. Area comparison of a 2-to-1 MUX using (a) one lateral six-terminal relay, and (b) two lateral four-terminal relays with connected drain and gate. The beam isolations regions are shown in red.

relay instead of two four-terminal relays because additional space for a second gate is required when using two four-terminal relays. For the lateral relays, the area of one six-terminal relay, $A_{6T\_lateral}$, is approximately $(8F + 2\gamma F)(5F + 2\gamma F + rF)$, where F is the minimum feature size, γ is the minimum anchor dimension in F, and r is the length-to-thickness ratio of the cantilever beam [Fig. 21(a)]. Assuming a timed isotropic etch process, γ is the minimum ratio between the width of the nonreleased regions (anchors) and the thickness of the released regions (Y-shaped cantilever beams) [1]. The area of two four-terminal relays, $A_{two\_4T\_lateral}$, is approximately (16F+2γF)(5F+2γF+rF) [Fig. 21(b)]. The typical ratio of $A_{6T\_lateral}$ to $A_{two\_4T\_lateral}$ ranges from 0.6 at $\gamma = 2$ to 0.67 at $\gamma = 4$.

Second, using a single gate ensures that there is no direct conductive path between two sources, i.e., between V_{DD} and GND, at any time during the gate voltage sweep. In contrast, if two four-terminal relays are used, there is the risk of simultaneously turning two relays on, which may cause a direct path between V_{DD} and GND.

Third, a single gate that has only two switching states does not suffer from the same settling related issue that occurs when using two gates [2]. As an illustration, suppose that one four-terminal relay is turned off and the next actuation signal is applied after a nominal switching delay. For a low damping system (quality factor $Q > 1$), the time for the gate to settle in its initial off-state, after it is turned off, is much

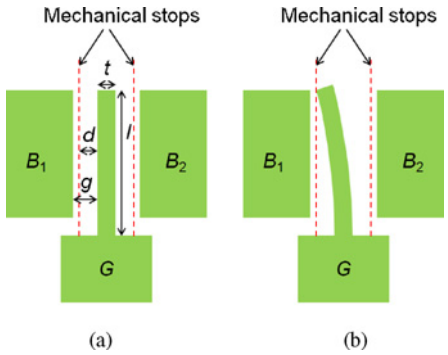


Fig. 22. Simplified model of a six-terminal relay for electrostatic analysis. (a) Initial nonactuated state. (b) Switching state to the left side. Assumptions include a full overlap between the body and the gate and a parabolic gate deflection. The source and drain electrodes are located at mechanical stops.

larger than the nominal switching delay. If the next actuation signal is applied while the gate is still in a settling state, then the switching delay can vary significantly, up to around twice of the nominal switching delay. This is because the initial state of the gate, i.e., the position and the velocity also affects the switching delay.

Fourth, using a single gate makes it easier for the gate to be released at pull-out than using two gates. For the six-terminal relay, the electrostatic force on the gate is exerted in two opposite directions by two body electrodes. The electrostatic force from the opposite body electrodes helps release the gate at pull-out, i.e., the sum of this electrostatic force and the elastic restoring force should be larger than the surface adhesion force.

APPENDIX III
DEVICE MODELING

To derive analytic expressions for first-order estimates of the device's mechanical behavior, we use a simple cantilever beam as shown in Fig. 22, where the contact areas between the gate and source, and the gate and drain are neglected. Also, we assume that there are only electrostatic and spring restoring forces. Then the pull-in voltage, resonant frequency (f_{res}), gate-to-body on-capacitance (C_{on}), and switching energy are given by or based upon [19]

$$V_{\text{PI}} = \sqrt{\frac{1.827E}{6\varepsilon_0}\frac{t^3g^3}{l^4}} \tag{6}$$

$$f_{\text{res}} = \frac{3.52}{2\pi}\sqrt{\frac{E}{12\rho}}\frac{t}{l^2} \tag{7}$$

$$C_{\text{on}} = \frac{\varepsilon_0 lh}{g} f_1\left(\frac{d}{g}\right) \tag{8}$$

$$E_{\text{sw}} = C_{\text{on}}V_{PI}^2 = \left(\frac{1.827E}{6}\right)\frac{g^2t^3h}{l^3} f_1\left(\frac{d}{g}\right) \tag{9}$$

where E is the Young's modulus of the beam material and t, h, and l are the thickness, height, and length of the cantilever beam, respectively. Also, g is the initial gate-to-body gap, d is the initial gate-to-drain (or source) gap, ε_0 is the permittivity of free space, ρ is the density of the beam, and the function, $f_1(x)$ is $\frac{1}{2\sqrt{x}}\left(\ln\left(1+\sqrt{x}\right) - \ln\left(1-\sqrt{x}\right)\right)$. The expressions are derived using a 2-D deflection model of the beam with a parabolic beam deflection approximation and assuming that the body and the gate have a full overlap [19]. As an example, we obtain $V_{\text{PI}} = 6.26$ V, $f_{\text{res}} = 8.61$ MHz, and $E_{\text{sw}} = 3.92$ fJ, where we use $t = 100$ nm, $h = 200$ nm, $l = 4\,\mu$m, $g = 120$ nm, $d = 100$ nm, $E = 169$ GPa, and $\rho = 2330$ kg/m^3 for a polysilicon beam. Note that V_{PI} and E_{sw} can decrease at the expense of decreasing f_{res} by increasing l.

With a 1-D lumped model, Newton's second law of the motion for a relay can be normalized to [20], [21]

$$\frac{d^2u}{d\tilde{t}^2} + \frac{1}{Q}\frac{du}{d\tilde{t}} + u = \frac{4}{27}\left(\frac{V_{DD}}{V_{PI}}\right)^2\frac{1}{(1-u)^2} \tag{10}$$

where u is the normalized displacement of the gate with $u = x/g$, $\tilde{t}$ is the normalized time with $\tilde{t} = \sqrt{k_{\text{eff}}/m_{\text{eff}}} \times t$, m_{eff} is the effective mass of the beam, and k_{eff} is the effective spring constant. Then, the switching delay is given by

$$t_{\text{sw}} = \tilde{t}_{\text{sw}}\sqrt{\frac{m_{\text{eff}}}{k_{\text{eff}}}}, \tag{11}$$

where $\tilde{t}_{\text{sw}}$ can be calculated from (10). For a six-terminal relay, $\tilde{t}_{\text{sw}}$ is the normalized time $(\tilde{t})$ that satisfies $u(\tilde{t}) = d/g$ with the initial boundary conditions: $u(0) = -d/g$ and $\frac{du}{d\tilde{t}}(0) = 0$.

When the surface adhesion forces are negligible, the relay performance can be improved by scaling all dimensions similar to the constant field scaling for MOSFETs (Table IV) [2]. Once the gap size reaches below 10–20 nm, the surface adhesion forces (in particular the van der Waals force) play a significant role. Note that with a scaling factor of $1/\kappa$, the scaling factor for the van der Waals is κ, while the scaling factor for the electrostatic and spring restoring force is $1/\kappa^2$. Therefore, it is necessary to decrease the beam aspect ratio, the length divided by the thickness, to ensure that the pull-out voltage is larger than zero, i.e., the spring restoring force is larger than the van der Waals force at pull-out, and does not decrease too rapidly with gap size to account for possible variation of the gap size due to fabrication variations [2].

TABLE IV
CONSTANT FIELD SCALING OF A RELAY

Relay Parameter	Scaling Factor
All dimensions (gap, thickness, height, length)	$1/\kappa$
Pull-in voltage	$1/\kappa$
Switching delay	$1/\kappa$
Switching energy	$1/\kappa^3$
Device area	$1/\kappa^2$
Electrostatic force	$1/\kappa^2$
Spring restoring force	$1/\kappa^2$
van der Waals force	κ

APPENDIX IV
DEVICE STRUCTURE COMPARISON: LATERAL VERSUS VERTICAL RELAYS

The relays fabricated in the plane of the substrate with isolated channels can be classified into two groups depending on the actuation direction: lateral relays and vertical relays. A

third group of device is possible based on a pillar fabricated out of the plane of the substrate and then actuated laterally [22]. For this analysis, we restrict our consideration to relays fabricate in the plane of the substrate because it is extremely challenging to extend the demonstrated two-terminal relay process for laterally actuated pillars to the relay process with isolated channels.

Lateral relays are suitable for implementing six-terminal relays since all electrodes and the gate can be patterned in a single lithographic step, which ensures the actuator symmetry and bi-directional actuation. In contrast, for vertical relays, typically a downward actuation is used only [6], [23]. An upward actuation can be performed only at the cost of a significant increase in the process complexity, i.e., stacking the second set of body, drain, and source electrodes on top of the structural layer of the gate. If such a structure can be fabricated, a six-terminal relay will have half of the area required for two four-terminal relays placed next to each other.

Another benefit of using the lateral relays compared to using the vertical relays is that the lateral relays suffer less from the bending of the movable gate due to the stress-gradient of the structural layer. This is because the structural layer is typically deposited normal to the wafer surface. Moreover, for the lateral relays, the bending of the movable gate due to the stress-gradient of the insulating and conducting layers may be canceled out due to the symmetry of the gate structure.

Demonstrated vertical relays with the gate isolation include four-terminal piston relays with folded springs, and seven-terminal see–saw relays [6], [23]. In the see–saw relays, despite only the downward actuation, the bi-directional rotations are achieved with a single gate by using torsional springs. The torsional springs should be designed such that the coupled bending of the springs can be negligible. These two relay structures have decoupled springs and movable gates so they take larger areas and are inherently slower due to larger mass of the movable gates compared to the lateral six-terminal relays. From [6] and [23], the four-terminal piston relay and the see–saw relay areas are estimated to be around $1500F^2$ and 4250F^2, respectively. The number is reduced to $528F^2$ at $r = 35$ and $\gamma = 2$ when using the lateral six-terminal relay.

Fig. 23 shows a schematic of a cantilever-based design for the vertical four-terminal relays, which is a variant of a similar structure proposed in [4], but neither has been demonstrated. The area of this four-terminal relay, $A_{4T\_vertical}$, is approximately 10F(3F+ γF + rF). The area is calculated to be $400F^2$ at $r = 35$ and $\gamma = 2$, which is larger than half of the area of one lateral six-terminal relay. Table V shows the logic gate area comparison for a 2-to-1 MUX implementation with typical values of γ and r.

As a comparison, the area of a MUX2_X1 cell from the 45 nm Nangate library is about $920F^2$ and the switching energy is calculated to be about 64.8 aJ [24]. The switching energy of a lateral six-terminal relay with the same area (F = 45 nm, γ = 2, $t = F$, $h = 2F$, $l = 67.7F$, $g = 1.2F$, $d = F$, E = 169 GPa, and ρ = 2330 kg/m^3) is calculated to be 73.6 aJ from (9). Therefore, the MOSFET and the same area lateral six-terminal relay have approximately the same switching energy at F = 45 nm.

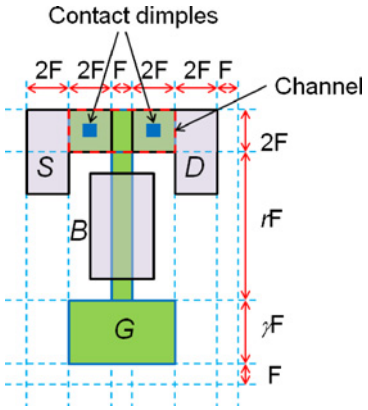


Fig. 23. Schematic of a layout for a vertical four-terminal relay with a cantilever beam and two contact dimples. Structural layer, contact, and electrodes for source, drain, and body are shown in green, blue, and bright purple, respectively. Channel is shown in a box with a dashed boundary in red.

TABLE V
AREA COMPARISON FOR A 2-TO-1 MUX IMPLEMENTATION WITH COMPACT CANTILEVER-BASED LOGIC GATE DESIGNS

Parameter Values	One Lateral Six-Terminal Relay (F^2)	Two Lateral Four-Terminal Relays (F^2)	Two Vertical Four-Terminal Relays (F^2)
γ = 2, r = 20	348	580	500
γ = 2, r = 35	528	880	800
γ = 4, r = 20	528	792	540
γ = 4, r = 35	768	1152	840

Scaling of the gap size and the beam thickness is critical in order to make the lateral relay process superior to the vertical relay process for fabricating a compact 2-to-1 MUX. Note that for the lateral relay process, the gap size and the beam thickness are set by the lithographic resolution, whereas for the vertical relay process, the gap size and the beam thickness are determined by deposited film thickness. Scaling for the lateral relay can be accomplished as we use advanced lithography tools in combination with applying techniques for obtaining sublithographic features such as spacer process [25]. This will bring the realizable gap size and beam thickness for the lateral relay process in line with those achievable for the vertical relay process.

ACKNOWLEDGMENT

The authors would also like to thank J.-O. Lee and Prof. J.-B. Yoon, KAIST, for depositing MOCVD TiN at the National Nanofab Center, Korea. They would also like to thank J. Hammer of e3 Software for his assistance in programming the PerlDD tool. All other fabrication for this paper was performed at the Stanford Nanofabrication Facility, a node of the National Nanofabrication Infrastructure Network funded by the NSF.

REFERENCES

[1] K. Akarvardar, D. Elata, R. Parsa, G. C. Wan, K. Yoo, J Provine, P. Peumans, R. T. Howe, and H.-S. P. Wong, "Design considerations for complementary nano-electro-mechanical logic gates," in *Proc. IEDM Tech. Dig.*, Dec. 2007, pp. 299–302.

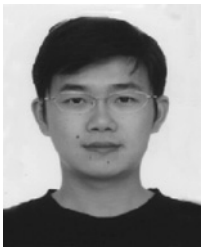
[2] K. Akarvardar and H.-S. P. Wong, "Nanoelectromechanical logic and memory devices," *ECS Trans.*, vol. 19, no. 1, pp. 49–59, 2009.
[3] V. Pott, H. Kam, R. Nathanael, J. Jeon, E. Alon, and T.-J. K. Liu, "Mechanical computing redux: Relays for integrated circuit applications," *Proc. IEEE*, vol. 98, no. 12, pp. 2076–2094, Dec. 2010.
[4] F. Chen, H. Kam, D. Markovic, T.-J. K. Liu, V. Stojanovic, and E. Alon, "Integrated circuit design with NEM relays," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2008, pp. 750–757.
[5] M. Spencer, F. Chen, C. C. Wang, R. Nathanael, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, D. Markovic, E. Alon, and V. Stojanovic, "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI application," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 308–320, Jan. 2011.
[6] R. Nathanael, V. Pott, H. Kam, J. Jeon, and T.-J. K. Liu, "Four-terminal relay technology for complementary logic," in *Proc. IEEE IEDM Tech. Dig.*, Dec. 2009, pp. 223–226.
[7] W. S. Lee, S. Chong, R. Parsa, J Provine, D. Lee, S. Mitra, H.-S. P. Wong, and R. T. Howe, "Dual sidewall lateral Nanoelectromechanical relays with beam isolation," in *Proc. 16th Int. Conf. Solid-State Sensors, Actuators Microsyst.*, Jun. 2011, pp. 2606–2609.
[8] C. E. Shannon, "A symbolic analysis of relay and switching circuits," *Trans. AIEE*, vol. 57, pp. 713–723, 1938.
[9] S. B. Akers, "Binary decision diagrams," *IEEE Trans. Comput.*, vol. 27, no. 6, pp. 509–516, Jun. 1978.
[10] F. Somenzi. (2012, May). "CUDD: CU decision diagram package," *Public Software* [Online]. Available: http://vlsi.colorado.edu/∼fabio/
[11] VIS Group, "VIS: A system for verification and synthesis," in *Proc. 8th Int. Conf. Comput.-Aided Verification*, LCNS 1102. Jul. 1996, pp. 428–432.
[12] K. Navi, O. Kavehie, M. Rouholamini, A. Sahafi, and S. Mehrabi, "A novel CMOS full adder," in *Proc. Int. Conf. VLSI Des.*, Jan. 2007, pp. 303–307.
[13] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1090, Jul. 1997.
[14] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. 35, no. 8, pp. 677–691, Aug. 1986.
[15] C. Yang, V. Singhal, and M. Ciesielski, "BDD decomposition for efficient logic synthesis," in *Proc. Int. Conf. Comput. Des.*, Oct. 1999, pp. 626–631.
[16] J. Earle, "Latched carry-save adder," *IBM Tech. Disclosure Bull.*, vol. 7, no. 10, pp. 909–910, Mar. 1965.
[17] D. Lee, W. S. Lee, J Provine, J.-O. Lee, J.-B. Yoon, R. T. Howe, S. Mitra, and H.-S. P. Wong, "Titanium nitridue sidewall stringer process for lateral nanoelectromechanical relays," in *Proc. 23rd IEEE Micro Electro Mech. Syst. Conf.*, Jan. 2010, pp. 456–459.
[18] Y. X. Liu, T. Kamei, K. Endo, S. O'uchi, J. Tsukada, H. Yamauchi, T. Hayashida, Y. Ishikawa, T. Matsukawa, K. Sakamoto, A. Ogura, and M. Masahara, "Nanoscale TiN wet etching and its application for FinFET fabrication," in *Proc. ISDRS*, Dec. 2009, pp. 1–2.
[19] G. T. A. Kovacs, *Micromachined Transducers Sourcebook*. New York: WCB/McGraw-Hill, 1998, pp. 278–281.
[20] D. Elata and H. Bamberger, "On the dynamic pull-in of electrostatic actuators with multiple degrees of freedom and multiple voltage sources," *J. Microelectromech. Syst.*, vol. 15, no. 1, pp. 131–140, Feb. 2006.
[21] H. Kam, T.-J. K. Liu, V. Stojanovic, D. Markovic, and E. Alon, "Design, optimization, and scaling of MEM relays for ultra-low-power digital logic," *IEEE Trans. Electron Devices*, vol. 58, no. 1, pp. 236–250, Jan. 2011.
[22] V. X. H. Leong, E. J. Ng, J. B. W. Soon, N. Singh, N. Shen, T. Myint, V. Pott, and J. M. Tsai, "Vertical silicon nano-pillar for non-volatile memory," in *Proc. 16th Int. Conf. Solid-State Sensors, Actuators Microsyst.*, Jun. 2011, pp. 649–652.
[23] J. Jeon, V. Pott, H. Kam, R. Nathanael, E. Alon, and T.-J. K. Liu, "Perfectly complementary relay design for digital logic applications," *IEEE Electron Device Lett.*, vol. 31, no. 4, pp. 371–373, Apr. 2010.
[24] Nangate. (2011, Aug.). *Nangate 45 nm Library* [Online]. Available: http://www.nangate.com
[25] D. Lee, S. Mitra, R. T. Howe, and H.-S. P. Wong, "Multi-spacer technique for low-voltage, high-aspect-ratio lateral electrostatic actuators," in *Proc. 16th Int. Conf. Solid-State Sensors, Actuators Microsyst.*, Jun. 2011, pp. 2602–2605.

Daesung Lee received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2007.

Since 2009, he has been a Post-Doctoral Researcher with the Department of Electrical Engineering, Stanford University. He has authored and co-authored more than 55 journal and conference papers. His current research interests include design and fabrication of scaled NEM relays, NEMS, and optical MEMS.

W. Scott Lee (S'12) received the B.S. and M.S. degrees in electrical engineering from the University of Utah, Salt Lake City, in 2008. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical Engineering, Stanford University, Stanford, CA.

His research is focused on designing high-speed interconnects for digital very large scale integration (VLSI) circuits in collaboration with Mentor Graphics, Wilsonville, OR. His work is focused on designing high-speed interconnects for digital VLSI circuits in collaboration with Mentor Graphics. His primary research interests include the fabrication and testing of NEM relays with an emphasis on developing NEM-relay-based systems.



Chen Chen received the B.Sc. degree in microelectronics from Peking University, Beijing, China, in 2008, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2010, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering since 2010.

In 2010, he held a three-month intern position with the Device Modeling Group, Altera Corporation at Altera Headquarters, San Jose, CA.

Mr. Chen was a recipient of the Fellowship from the Department of Electrical Engineering, Stanford University, and the Top-Class Scholarship from Peking University.

Farzan Fallah received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1999.

He is currently a Scientist with the Department of Electrical Engineering, Stanford University, Stanford, CA. After receiving the Ph.D. degree, he joined the Fujitsu Labs of America, Sunnyvale, CA, where he led the Low-Power Design Project until October 2008. He has published more than 70 papers and has 22 patents granted or pending.

Dr. Fallah was a recipient of the Best Paper Award at the Design Automation Conference in 1998 and the Best Paper Award at the VLSI Design Conference in 2005.

J Provine (M'06) received the B.A. degree in physics in 1998, the B.S. and M.S. degrees in electrical engineering all from Rice University, Houston, TX, in 1999, and the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY, in 2005.

He is currently a Research Associate with the Department of Electrical Engineering, Stanford University, Stanford, CA. His current research interests include surface-mode photonics, NEMS, and surface force probes.

Soogine Chong (S'07) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 2006 and 2012, respectively.

She is currently with Samsung Electronics, Seoul. She was a summer Intern with the Interuniversity Microelectronics Center, Leuven, Belgium, in 2006, and with IBM, Fishkill, NY, in 2007. Her current research interests include the design and fabrication of nanoelectromechanical relays integrated with CMOS.

Dr. Chong was a recipient of the Samsung Scholarship from 2004 to 2009.

John Watkins (M'01) received the B.S. and M.S. degrees in electrical engineering from Stanford University, Stanford, CA, in 1978.

He has done computer systems development at Hewlett-Packard, Sun Microsystems, and Oracle Corporation with products in the server, workstation, and I/O subsystems areas. He is currently a Consulting Technical Staff Member with Oracle Labs, Belmont, CA. He is an inventor of more than 25 granted patents.

Roger T. Howe (S'80–M'84–SM'93–F'96) received the B.S. degree in physics from Harvey Mudd College, Claremont, CA, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley (UC Berkeley), in 1981 and 1984, respectively.

He is currently the William E. Ayer Professor with the Department of Electrical Engineering, Stanford University, Stanford, CA. From 1984 to 1987, he was a Faculty Member with Carnegie Mellon University, Pittsburgh, PA, and with the Massachusetts Institute of Technology, Cambridge. He returned to UC Berkeley in 1987, where he remained on the Faculty until 2005. He was a co-founder of Silicon Clocks, Inc., a start-up company commercializing integrated MEMS resonator-based timing products which was acquired by Silicon Laboratories, Inc., in 2010. In 2011, he became the Director of the National Nanotechnology Infrastructure Network. His current research interests include the applications of micro- and nanomachining to thermal energy conversion, microfluidics, and biomolecular sensing, as well as NEM relays.

Dr. Howe was a co-recipient of the 1998 IEEE Cledo Brunetti Award and was elected to the U.S. National Academy of Engineering in 2005 for his contributions to MEMS processes, devices, and systems. He is an editor of the *Journal of Microelectromechanical Systems*.

H.-S. Philip Wong (S'81–M'82–SM'95–F'01) received the B.Sc. degree (Hons.) from the University of Hong Kong, Pokfulam, Hong Kong, in 1982, the M.S. degree from the State University of New York, Stony Brook, in 1983, and the Ph.D. degree from Lehigh University, Bethlehem, PA, in 1988, all in electrical engineering.

He was with IBM Research, T. J. Watson Research Center, New York, for 16 years, where he worked on charged-couple device and CMOS image sensors, double-gate/multigate MOSFET, device simulations for advanced/novel MOSFET, strained silicon, wafer bonding, ultrathin-body silicon-on-insulator (SOI), extremely short-gate field-effect transistors (FETs), germanium MOSFET, carbon nanotube FET, and phase change memory. He has held various positions ranging from a Research Staff Member to a Manager and a Senior Manager. While he was the Senior Manager, he had the responsibility of shaping and executing IBM's strategy on nanoscale science and technology, as well as exploratory silicon devices and semiconductor technology. In 2004, he joined the Department of Electrical Engineering, Stanford University, Stanford, CA. His current research interests include nanoscale science and technology, semiconductor technology, solid-state devices, electronic imaging, exploring new materials, novel fabrication techniques, and novel device concepts for future nanoelectronic systems.

Dr. Wong is a member of the Emerging Research Devices Working Group of the International Technology Roadmap for Semiconductors. From 2001 to 2006, he served on the IEEE Electron Devices Society as an Elected AdCom Member. From 1998 to 2007, he served on the International Electron Devices Meeting (IEDM) Committee and was the Technical Program Chair in 2006 and the General Chair in 2007. From 1998 to 2004, he served on the International Solid State Circuits Conference (ISSCC) Program Committee and was the Chair of the Image Sensors, Displays, and Microelectromechanical Systems subcommittee from 2003 to 2004. He currently serves on the Executive Committee of the Symposia of Very Large-Scale Integration (VLSI) Technology and Circuits. From 2005 to 2006, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON NANOTECHNOLOGY. He is also a Distinguished Lecturer of the IEEE Electron Devices Society and the IEEE Solid-State Circuits Society. He has taught several short courses at the IEDM, ISSCC, Symposium VLSI Technology, Silicon-on-Insulator Conference.

Subhasish Mitra (SM'06–F'13) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He is currently an Associate Professor with the Departments of Electrical Engineering and Computer Science, Stanford University, where he leads the Stanford Robust Systems Group and is a Chambers Faculty Scholar of engineering. Prior to joining Stanford University, he was a Principal Engineer with Intel Corporation. His X-compact technique for test compression has been used in more than 50 Intel products, and has influenced major electronic design automation tools. His current research interests include robust system design, very large scale integration (VLSI) design, computer-aided design, validation and testing, and emerging nanotechnologies.

Dr. Mitra's major honors include the Presidential Early Career Award for Scientists and Engineers (PECASE, the highest honor bestowed by the U.S. government on early-career outstanding scientists and engineers) from the White House, the Terman Fellowship, the IEEE Circuits and Systems (CAS)/Council on Electronic Design Automation (CEDA) Donald O. Pederson Award for the Best Paper published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the Intel Achievement Award, Intel's highest corporate honor. He and his students presented award-winning papers at several major conferences, including the IEEE/ACM Design Automation Conference, the IEEE International Test Conference, the IEEE VLSI Test Symposium, the Intel Design and Test Technology Conference, and the Symposium on VLSI Technology. At Stanford University, he was honored several times by graduating seniors for being important to them during their time at Stanford University. He has served on numerous conference committees and journal editorial boards. Recently, he served on the Defense Advanced Research Projects Agency's Information Science and Technology Board as an Invited Member. The IFRA technology for post-silicon validation, created jointly with his student, was characterized as a breakthrough in a Research Highlight in the Communications of the ACM. His work on the first demonstration of carbon nanotube imperfection-immune digital VLSI, jointly with his students and collaborators, was selected by the National Science Foundation as a Research Highlight to the United States Congress, and was highlighted as a significant breakthrough by the Semiconductor Research Corporation, the MIT Technology Review, the New York Times, and several others.