

Embedded Nano-Electro-Mechanical Memory for Reconfigurable Lookup Tables

Kimihiko Kato, Vladimir Stojanović and Tsu-Jae King Liu

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA
Phone: +1-510-642-2689 E-mail: k.kato@eecs.berkeley.edu, tking@eecs.berkeley.edu

Abstract

A novel reconfigurable lookup table (LUT) architecture implemented using embedded non-volatile nano-electro-mechanical switches (NEMory) is proposed for fast and energy-efficient computing.

1. Introduction

The future Internet of Things (IoT) will require embedded electronics to perform real-time computation on large sets of data with high energy-efficiency. A super-parallel computing scheme implemented using a non-volatile memory (NVM) array with simple arithmetic functionality, called in-memory computing (IMC), has been proposed recently to address this need [1]. Non-volatile nanometer-scale electro-mechanical memory (NEMory) switches are ideally suited for IMC due to their zero standby power [2] and lower programming energy than other NVM devices [3].

In this paper, a new circuit architecture and operating scheme is proposed for faster and more energy-efficient implementation of lookup tables (LUTs) with NEMory than with either CMOS or ReRAM devices (**Fig. 1**). Vertically oriented non-volatile NEM switches implemented using the back-end-of-line (BEOL) air-gapped interconnect layers of an advanced CMOS process [4] provide for a compact NEMory cell layout and reprogrammability (**Fig. 2**): the movable electrode is physically anchored to a bottom bit-line (BL) layer, actuation electrodes (PL0 and PL1) on either side of the movable electrode are implemented in intermediate layers, and the contacting electrodes (I/O0 and I/O1) are implemented in a top input/output electrode layer.

2. LUT circuit architecture and operating scheme

The proposed circuit consists of a cross-point NEMory array with an input (Address) portion and an output (Result) portion, as illustrated in **Fig. 3** for a 5-input/2-output LUT; the BLs in the input portion are connected to the BLs in the output portion via CMOS pass gates. The number of columns in the NEMory array is $N+M$, where N is the number of input bits and M is the number of output bits. A set of PL0 and PL1 actuation electrode lines (not shown here for clarity) is shared across the cells within a single column, and is used to set the state of each NEM switch via electrostatic actuation to bring it into physical contact with either an I/O0 electrode or an I/O1 electrode. A non-linear device is assumed to be integrated into each switch, either at the bottom (Schottky contact) or at the top (metal-insulator-metal contact), to prevent sneak leakage paths in the cross-point array.

The number of rows in the NEMory array corresponds to the number of possible input bit combinations (up to 2^N); each input bit combination and its corresponding answer is programmed in the input portion and output portion, respectively, as follows (**Fig. 2c**): one BL is grounded at a time to program the cells one row at a time; a programming

voltage pulse (V_{prog}) is applied to the PL0 electrode in each column in which the cell is to be set to the “0” state; then V_{prog} is applied to the PL1 electrode in the other columns to set the remaining cells to the “1” state. Note that the input/output electrodes are electrically floating during a program operation, so that no direct current flows (*i.e.* the NEMory cell is “cold-switched”). The spring restoring force (F_{spring}) of the movable electrode should be weaker than the contact adhesive force (F_{adh}) so that contact is maintained with no applied voltage, *i.e.* for NV storage.

A lookup operation involves 3 steps: (1) with the CMOS pass gates disabled (Clock = GND), the bit lines are all precharged low (to GND) in the input portion of the array and the output lines are all precharged high (to V_{DD}) in the output portion of the array; (2) the input lines are driven, causing all but one bit line – that is, the one corresponding to the input bit combination – to be charged high, as indicated by the arrows in **Fig. 3a**; (3) the CMOS pass gates are enabled (Clock = V_{DD}), causing one of each pair of output lines to discharge toward GND according to the states of the cells connected to the one bit line that remained low, as indicated by the dotted arrows in **Fig. 3a**, so that the result can be detected by differential sense amplifiers (one for each output bit). It should be noted that, in principle, the electrical connections in the top electrode layer can be hardwired (vs. programmed) in the input portion of the array if there is no need for customization or reconfigurability.

3. NEMory LUT performance

The reprogramming characteristics of a NEMory cell are investigated herein via three-dimensional device simulations using Coventor MEMS+. **Fig. 4** shows the simulated structure; the actuation gap is assumed to correspond to the minimum feature size (F), while the contact gap ($F/2$) is assumed to be formed by a sub-lithographic patterning technique such as tilted ion implantation [5]. The minimum reprogramming voltage is plotted in **Fig. 5**, in the range $F_{\text{adh}} > F_{\text{spring}}$. Transient simulations (**Fig. 6**) show that a NEMory cell can be reprogrammed in less than 5 ns. Smaller gap size is beneficial for lower programming energy and delay (**Fig. 7**).

Depending on the contact resistance (R_{cont}), which limits the rate at which a bit line is charged and the rate at which an output line is discharged, stored answers can be looked up in less than 1 ns (no matter the number of output bits) using less than 1 pJ of energy for 4 output bits (**Fig. 8**). Performance characteristics of NEMory-based LUT (for 1 output bit) are benchmarked against those of ReRAM- and CMOS-based LUTs in **Fig. 9**.

4. Conclusion

A compact embedded NEMory array implemented using air-gapped interconnect layers can function as a lookup table with much better speed and energy efficiency than CMOS or ReRAM implementations. Also, it can be reprogrammed with less than 10 V for reconfigurable functionality.

Acknowledgements

This work was supported in part by the Center for Energy Efficient Electronics Science (NSF Award 0939514). K. Kato gratefully acknowledges support by a post-doctoral fellowship from JSPS.

References

- [1] B. Chen *et al.*, 2015 IEEE IEDM, 17.5.1.
- [2] K. Kato *et al.*, IEEE EDL **37**, 31 (2016).
- [3] N. Xuo *et al.*, 2014 IEEE IEDM, 28.8.1.
- [4] S. Natarajan *et al.*, 2014 IEEE IEDM, 3.7.1.
- [5] S. W. Kim *et al.*, Proc. SPIE 97771B (2016).
- [6] S. Paul *et al.*, "Computing with Memory for Energy-Efficient Robust Systems," Springer, 2014.
- [7] B. Govoreanu *et al.*, 2011 IEEE IEDM, 31.6.1.

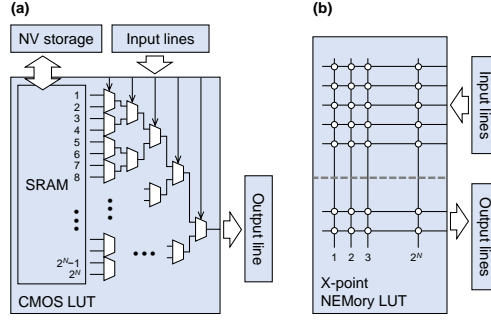


Fig. 1. Circuit block-level comparison of LUT implementations (a) CMOS based, and (b) NEMory based.

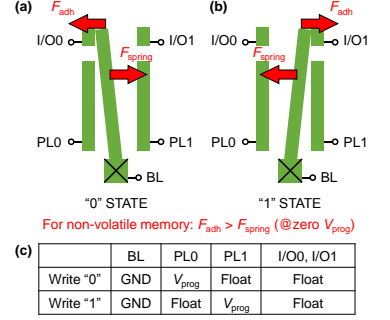


Fig. 2. Schematic illustration of NEMory switch in (a) "0" state and (b) "1" state. (c) programming voltage conditions.

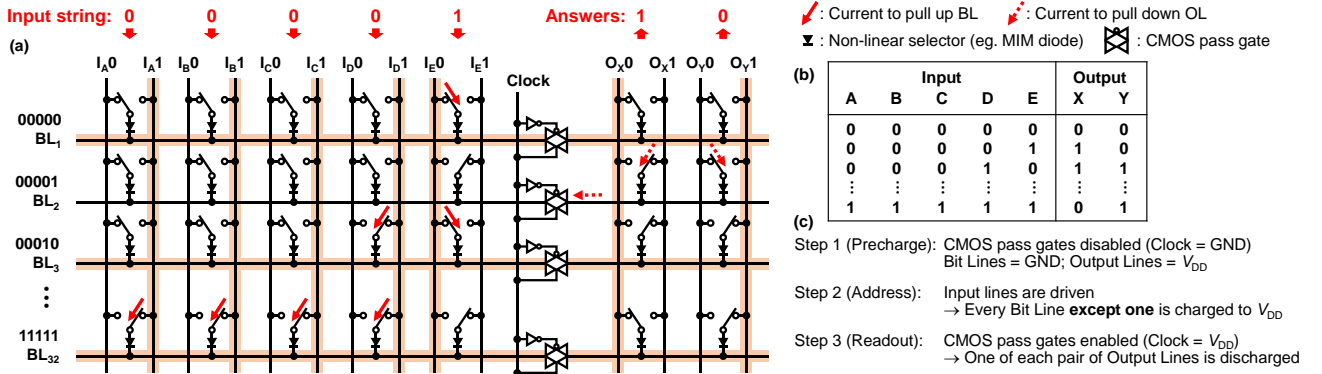


Fig. 3. (a) Circuit schematic for (5-input and 2-output) NEMory-based LUT, (b) corresponding truth table, and (c) readout scheme. Highlighted lines indicate driven wires (V_{DD} applied) for an input string of 00001 and answers 1 and 0, as an example.

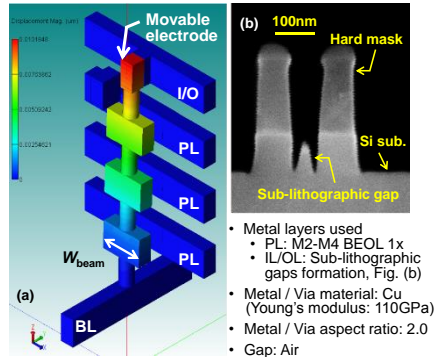


Fig. 4. (a) Simulated 3-D NEMory cell structure. (b) Demonstration of sub-lithographic (contact) gaps formed by tilted ion implantation [5].

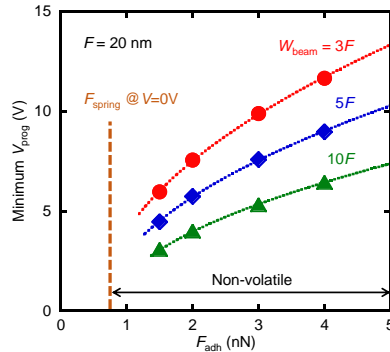


Fig. 5. Minimum voltage required to reprogram a NEMory cell, as a function of F_{adh} , for various W_{beam} .

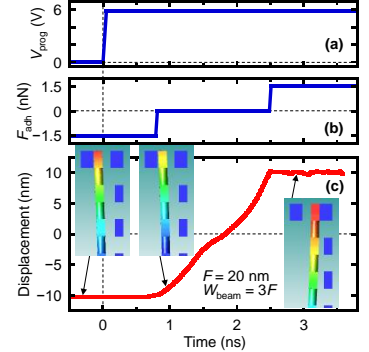


Fig. 6. Transient simulation of NEMory cell reprogramming: (a) V_{prog} , (b) F_{adh} , and (c) beam displacement (top layer).

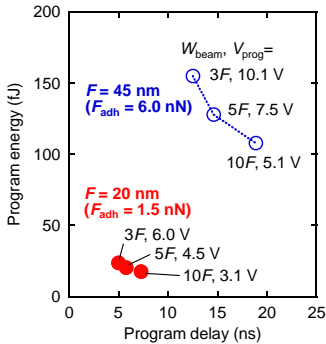


Fig. 7. Energy vs. delay tradeoff for programming a NEMory cell. W_{beam} is varied from $3F$ to $10F$, where F is the minimum feature size.

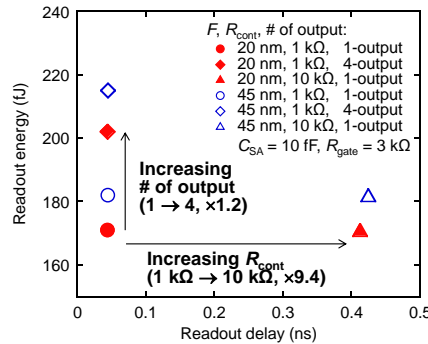


Fig. 8. Energy vs. delay tradeoff for lookup operation of a NEMory array. $W_{beam} = 3F$ and R_{cont} is the contact resistance.

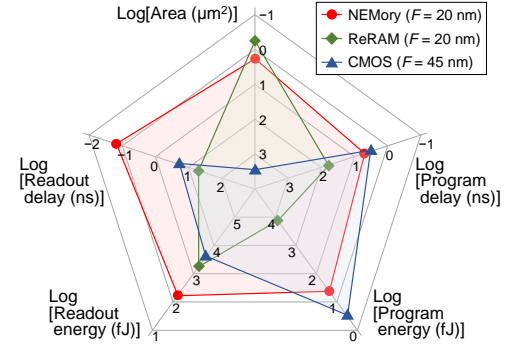


Fig. 9. Radar plot comparing the proposed NEMory-based LUT vs. ReRAM-based [1,7] and CMOS-based [6] LUTs.