Reconfigurable Memristor Crossbar for Graphlet Computing

Kyung Seok Woo[1,2,3], Nestor Ghenzi[4], Hyungjun Park[4], A. Alec Talin[1], Cheol Seong Hwang[4], R. Stanley Williams[1,2], and Suhas Kumar[1]

[1] *Sandia National Laboratories, Livermore, CA, USA*

[2] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA*

[3] *Advanced Light Source, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

[4] *Department of Materials Science and Engineering and Inter-university Semiconductor Research Center, Seoul National University, Seoul, Republic of Korea*

E-mail: kevinwoo@tamu.edu

In-memory computing approaches using memristor crossbars have been proposed as a paradigm shift to overcome the von Neumann bottleneck by combining memory and processing functions. Memristors are considered one of the most promising devices for in-memory computation due to their low power consumption, high switching speed, and scalability. However, sneak paths and stochastic behavior are two critical issues that limit their practical implementation. There have been transformative designs of in-memory computing using either stochasticity or sneak paths, but the combined potential of memristor crossbars remains unexplored. Here we propose a graphlet computing platform that synergistically utilizes both phenomena using a tunable hybrid memristor with a bilayer of insulator fabricated via plasma-enhanced atomic layer deposition (PEALD). By controlling the O_2 plasma power of PEALD, two oxide layers have different oxygen vacancy concentrations that allow for tunable switching behaviors under different switching conditions (Fig. 1). The tunable memristor was integrated into a crossbar, which was used to map a graph network by assigning each device as either a node or an edge. Then, we performed graphlet computing by utilizing inherent sneak paths and stochastic behavior of the crossbar (Fig. 2). While the sneak paths are used to count graphlet structures for analyzing complex graphs, the stochasticity is implemented in a random walk process to efficiently solve computationally expensive problems. This newly proposed computing scheme demonstrates the advancement of memristor-based in-memory computing hardware by addressing the inherent issues of memristor technology.

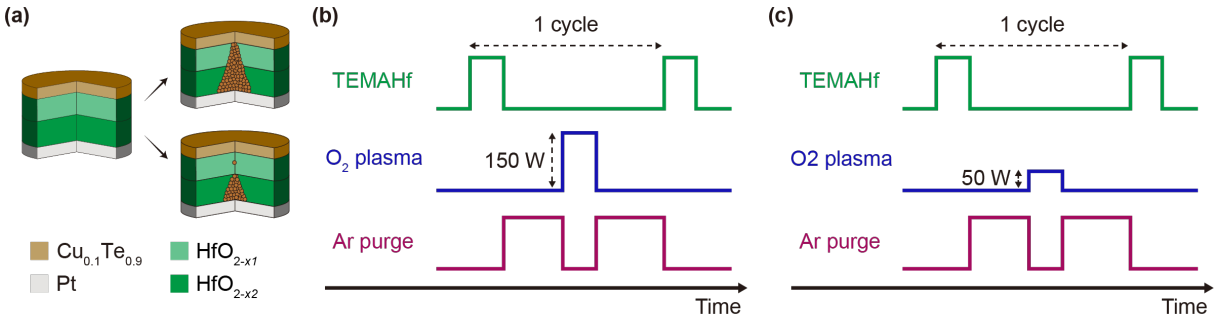

Fig 1. a) Device structure with two switching modes. ALD pulse sequences for b) HfO_{2-x1} and c) HfO_{2-x2}.

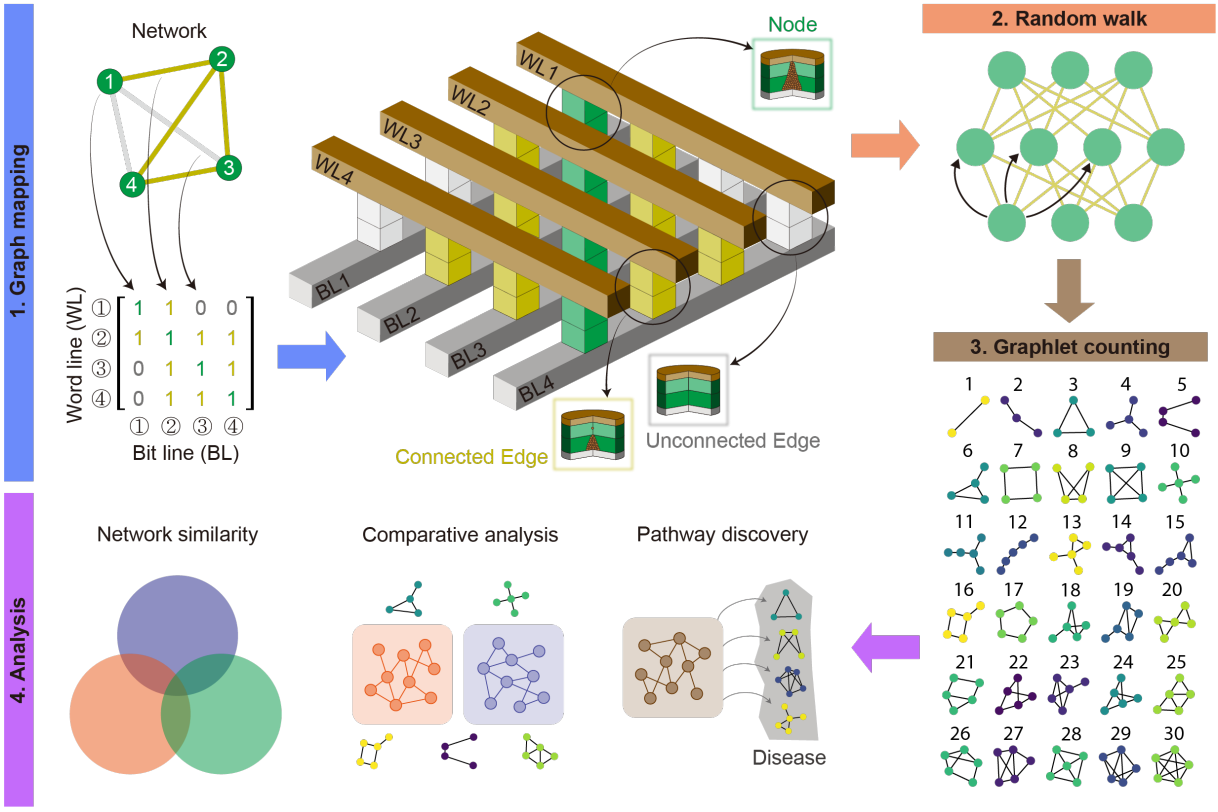

Fig 2. Memristor-based graph computing framework.