Balanced MAGIC: Impeding power side-channel attacks by masking data dependency of memristor operations

Lorenzo Pfeifer, Melvin Galicia, Rainer Leupers

RWTH Aachen, Germany

Memristors have become an important post-CMOS technology that enables new computing paradigms such as logic-in-memory with MAGIC. With MAGIC, binary switching states are used to implement basic logic functions that offer a promising alternative to conventional FPGAs with better energy efficiency[1]. However, these new processing paradigms come with new hardware security challenges. One such attack is EXAMINER[2], which exploits data-dependent power usage of memristors to extract sensitive intellectual property (IP).

This work proposes a countermeasure against such power side-channel attacks on MAGIC memristor crossbar logic. It works by masking the power usage with a second crossbar that computes an inverted replica of the original Boolean function. Each gate in this complementary network receives inverted inputs and produces an inverted output, so the function is evaluated once in normal and once in negated form. Thus every operation causes at least one power spike, either in the original or in the inverted network. This masks the data dependence of switching activity and impedes extraction of the implemented IP.

To implement this scheme, the inverse of all MAGIC gates must be available. MAGIC natively supports NOR and NOT gates. By De Morgan's law, the inverse of NOR is NAND, while NOT is self-inverse. The second crossbar must therefore implement NOT and NAND. Although NAND usually cannot be realized because it requires two serial paths, we propose a special 4T1R crossbar with a second connection parallel to the SelectLine, as shown in Fig. 1.

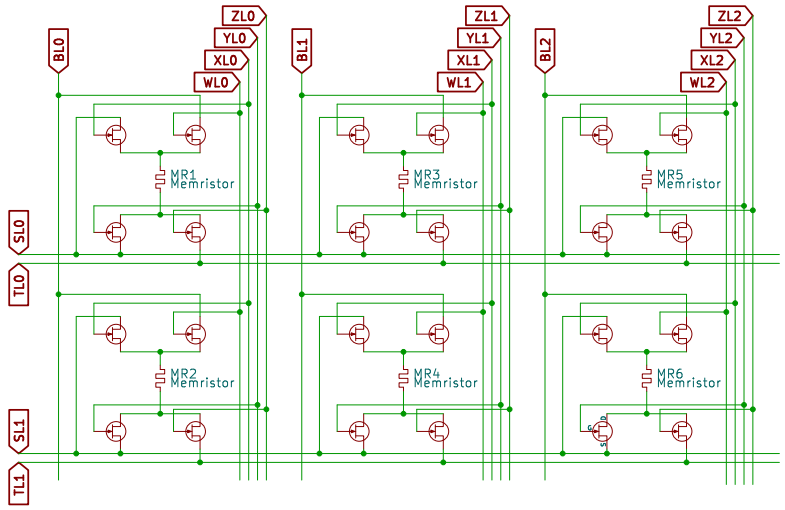

Figure 1: Schematic of proposed complementary 4T1R crossbar

[1] Kvatinsky et al., IEEE Transactions on Circuits and Systems II: Express Briefs, MAGIC—Memristor-aided logic, 61(11), pp. 895-899, 2014

[2] Pfeifer et al., IEEE Computer Society Annual Symposium on VLSI (ISVLSI), EXAMINER: IP Extraction Algorithm from MAGIC Logic-in-Memory, pp. 1-6, 2025